

Implementace technologie ZIGBEE do systému inteligentního domu

Implemenatation ZIGBEE Technology in Intelligent House

Zadání bakalářské práce

Student:

Jaromír Dudek

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R059 Mobilní technologie

Téma:

Implementace technologie ZIGBEE do systému inteligentního domu
Implemenatation ZIGBEE Technology in Intelligent House

Zásady pro vypracování:

Cílem práce je implementace bezdrátové technologie ZIGBEE do systému iDům (systému inteligentního domu vyvíjeného na katedře Telekomunikační techniky). Důležitá je realizace vybraných součástí systému (splnění bodu 3 a 4).

1. Uveďte výhody a nevýhody technologie ZIGBEE.
2. Popište komunikační protokol s rozhraním modulu ZIGBEE.
3. Realizujte rozhraní ZIGBEE - iDům.
4. Realizujte ZIGBEE komponenty systému iDům: vypínač, analogové čidlo, informační prvek s displejem.
5. Zhodnoťte systém z hlediska bezpečnosti přenášovaných dat.

Seznam doporučené odborné literatury:

EADY, Fred. Hands-On ZigBee: Implementing 802.15.4 with Microcontrollers. Burlington, Elsevier, 2007, ISBN: 0-1237-0887-7.

GISLASON, Drew. Zigbee Wireless Networking. Burlington, Elsevier, 2008, ISBN: 978-0-7506-8597-9.
Podle zadání vedoucího bakalářské práce.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Michal Jahelka, Ph.D.**

Datum zadání: 19.11.2010

Datum odevzdání: 06.05.2011



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 6. května 2011

.....

Rád bych na tomto místě poděkoval hlavně svému vedoucímu bakalářské práce – panu Ing. Michalu Jahelkovi, Ph.D. za trpělivost, cenné rady a zapůjčení vývojových kitů firmy Microchip. Dále pak panu doc. Mgr. Jiřímu Dvorskému, Ph.D. za perfektně zpracovanou šablonu `diploma.cls` pro \LaTeX .

Abstrakt

Bakalářská práce se zabývá implementací bezdrátové technologie ZigBee do systému inteligentního domu s názvem iDům, který je vyvíjen na katedře telekomunikační techniky Vysoké školy Báňské ve spolupráci s firmou ELEKTRO-FA. PAVELEK, s.r.o. Cílem bylo vyvinout prototyp modulu, který by zprostředkovával komunikaci mezi systémem iDům a koncovými bezdrátovými moduly. Dalším cílem bylo vyvinout prototypy bezdrátových modulů.

Klíčová slova: inteligentní dům, iDům, ZigBee, Bezdrátové technologie, IEEE 802.15.4

Abstract

The aim of my bachelor's thesis was to implement wireless technology – ZigBee into intelligent house system iDům which are development at VŠB-Technical University of Ostrava on Department of Telecommunications in cooperation with the ELEKTRO-FA. PAVELEK, s.r.o. The first objective was to develop prototype, which translate messages from iDům system into wireless modules. Next objective was to develop prototypes of wireless end devices.

Keywords: intelligent home automation, iDům, ZigBee, Wireless, IEEE 802.15.4

Seznam použitých zkratk a symbolů

AES	– Advanced Encryption Standard
EEPROM	– Electronic Erasable Programable Read Only Memory
FHSS	– Frequency Hopping Spread Spectrum
ISM	– Industrial, Scientific and Medical (Pásmo rádiového vlnění, frekvence cca 2,5 GHz)
LCD	– Liquid Crystal Display
MAC	– Media Access Control, Media Access Control Address (adresa síťového rozhraní)
OUI	– Organizationally Unique Identifier
PTMP	– Point To MultiPoint
PTP	– Point To Point
SPI	– Serial Peripheral Interface
UART	– Universal Asynchronous Receiver/Transmitter

Obsah

1	Úvod	5
1.1	Stanovené cíle	5
2	Bezdrátové technologie	6
2.1	Bluetooth	6
2.2	WiFi	7
2.3	IQRF	8
2.4	ZigBee	9
2.5	MiWi(TM)	11
2.6	Srovnání technologií	12
3	Technologie iDům	14
3.1	Popis systému iDům	14
3.2	Komunikace	15
4	Vývoj	16
4.1	Programování a ladění modulů	16
4.2	Microchip ZigBee stack 2006	17
4.3	Konfigurace bezdrátových zařízení	20
5	Vyvíjené moduly	24
5.1	Vlastnosti společné všem modulům	24
5.2	Modul rozhraní iDům - ZigBee	27
5.3	Modul vypínač	30
5.4	Modul LCD	33
6	Závěr	35
7	Reference	36
	Přílohy	37
A	Výpisy kódů	38
B	Vývojové diagramy	43
C	Osazovací plánky	44
D	Schémata zapojení	45

Seznam tabulek

1	Přehled standardů 802.11 [14]	8
2	Porovnání bezdrátových technologií dle parametrů	12
3	Formát iDům zprávy [4]	15
4	Porovnání bezpečnostních režimů[8]	21
5	Význam jednotlivých pinů na desetipinovém dvouřadém konektoru . . .	24

Seznam obrázků

1	ZigBee topologie	10
2	MiWi(TM) vývojové prostředí	12
3	Vývojový kit explorer16	16
4	Propojovací modul	16
5	Aplikace ZENA a její záložky	23
6	Komunikace s modulem pomocí sériového portu	25
7	RF modul MFR24J40MA	25
8	Blokový diagram modulu rozhraní iDům – ZigBee	27
9	Znázornění funkce modulu rozhraní iDům – ZigBee	28
10	Plošný spoj modulu rozhraní iDům – ZigBee	29
11	Hotový modul rozhraní iDům – ZigBee	29
12	Blokový diagram modulu vypínač	31
13	Plošný spoj modulu vypínač	32
14	Hotový modul vypínač	32
15	Blokový diagram modulu LCD	33
16	Plošný spoj modulu s LCD displayem	34
17	Hotový modul LCD	34
18	Zpracování iDům zprávy	43
19	Osazovací plány modulů	44
20	Schéma modulu rozhraní iDům – ZigBee	45
21	Schéma modulu vypínač	46
22	Schéma modulu s LCD displayem	47
23	Schéma propojovacího modulu	48

Seznam výpisů zdrojového kódu

1	Základní struktura programu	17
2	Ukázka příjmu ZigBee zprávy	18
3	Ukázka odeslání ZigBee zprávy	19
4	Ukázka odeslání zabezpečené ZigBee zprávy	20
5	Struktura <code>wirelessDevInfo</code>	27
6	Odeslání zprávy o změně stavu	30
7	Zpracování přijaté zprávy od systému iDům	38
8	Zpracování přijaté zprávy od bezdrátového zařízení	39
9	Ukázka souboru <code>zigbee.def</code>	39
10	Ukázka souboru <code>myZigBee.c</code>	41

1 Úvod

Tato bakalářská práce pojednává o návrhu a výrobě prototypů bezdrátových modulů, určených pro systém inteligentního domu s názvem iDům, který je vyvíjen na katedře telekomunikační techniky Vysoké školy Báňské ve spolupráci s firmou ELEKTRO-FA. PAVELEK, s.r.o.

Na začátku bakalářské práce je prostor věnován seznámení s bezdrátovými technologiemi a výběru vhodné technologie, dále pak samotnému vývoji. Na konci je část s přílohami, kde se nacházejí obrázky, tabulky a výpisy kódů.

1.1 Stanovené cíle

- Cílem bakalářské práce je vyvinout prototyp bezdrátového modulu „rozhraní iDům – ZigBee“, který by zprostředkoval komunikaci mezi systémem inteligentního domu iDům a bezdrátovými moduly, založenými na technologii 802.15.4, tzv. ZigBee.
- Dalším cílem je vytvořit prototypy bezdrátových bateriově napájených modulů, které budou pomocí modulu iDům – ZigBee připojeny do systému inteligentního domu iDům
- Celý systém by měl být pokud možno co nejvíce transparentní a bez úprav stávajícího systému iDům.

2 Bezdrátové technologie

2.1 Bluetooth

Bluetooth je bezdrátová technologie založená na standardu 802.15.1. Specifikace byla vyvinuta v roce 1994 Jaapem Haartsenem a Svenem Mattissonem, kteří pracovali pro firmu Ericsson ve Švédském Lundu. Bluetooth pracuje v ISM pásmu. K přenosu informací se využívá metoda FHSS, která by měla zvýšit odolnost spojení vůči rušení na stejné frekvenci. Je definováno několik výkonových úrovní:

- 1 mW, dosah 1 metr,
- 2,5 mW, dosah cca 10 metrů,
- 100 mW, dosah cca 100 metrů.

Standard Bluetooth je definován v několika verzích, v současné době je asi nejvíce využívána verze 2.0. Technologie Bluetooth je hojně využíváno v mobilních telefonech pro připojení různých handsfree a dále v notebookech pro pohodlný přenos dat.[3]

Verze Bluetooth:

- Verze 1.0 a 1.b
 - výrobci měli mnoho problémů, aby byly jejich výrobky interoperabilní.
- Verze 1.1
 - v roce 2002 schváleno jako standard,
 - přidána podpora pro nešifrované kanály,
 - indikátor síly signálu tzv RSSI (Received Signal Strength Indication).
- Verze 1.2
 - 1 Mbit/s,
 - zpětně kompatibilní s 1.1,
 - rychlejší připojení a vyhledávání zařízení,
 - vyšší přenosové rychlosti než v 1.1.
- Verze 2.1 + EDR, 3 Mbit/s
 - plně zpětně kompatibilní s 1.2,
 - hlavním rysem je bezpečné jednoduché párování (SSP).
- Verze 3.0 + HS, 24 Mbit/s
- Verze 4.0, 24 Mbit/s

Jednotlivá zařízení jsou identifikovatelná pomocí své BD_ADDR (Bluetooth Device Address), podobně jako např. MAC u ethernetu. Bluetooth podporuje komunikaci typu PTP i PTMP. První stanice, která se připojí do sítě, je tzv. Master. Úkolem Mastera je řídit komunikaci s ostatními stanicemi, tzv. SLAVE, kterých může být maximálně sedm.[3]

Bezpečnost Bluetooth je řešena pomocí několika úrovní:

- **Párování** – před prvním přenosem dat se provádí „párování“, které propojí dvě konkrétní zařízení pomocí bezpečnostního kódu tzv. PIN.
- **Skrývání zařízení** – zařízení není možné vyhledat pomocí standardního protokolu SDP (Service Discovery Protocol), to však neznamená, že se k zařízení není možné připojit.
- **Šifrování komunikace** – pomocí metody AES-128 bitů.

2.2 WiFi

WiFi (Wireless Fidelity) je bezdrátová technologie založená na standardu 802.11., který vznikl v roce 1997. V roce 1999 se standard rozšířil o dvě kvalitnější specifikace, 802.11a a 802.11b. Nejpoužívanější je specifikace 802.11g. Původně bylo WiFi určeno pro vzájemné bezdrátové propojení přenosných zařízení a jejich připojování do sítí typu LAN (Local Area Network). V dnešní době je hojně využívána jako technologie poslední míle pro připojení domácnosti k internetu a na veřejnosti tzv. Hotspotů.[14]

Typy sítí:

- **Ad-Hoc** – přímé spojení koncových zařízení, není třeba žádný prostředník, pouze pro malé sítě.
- **Infrastruktura** – v síti je hlavním uzlem přístupový bod (AP – Access Point), který řídí komunikaci. Vysílá SSID (Service Set Identifier) identifikátor sítě všem potenciálním klientům. Klient si pak následně podle SSID zvolí, ke které síti se bude připojovat.

2.2.1 Zabezpečení WiFi sítí

- První a základní možnost je skrývání SSID – klient síť nevidí v seznamu sítí, proto jako by pro něj neexistovala.
- Kontrola MAC adresy Access Point si udržuje seznam připojených klientů, kterým je dovoleno se do sítě připojit.
- Šifrování komunikace pomocí statických klíčů WEP (Wired Equivalent Privacy) symetrické šifry, které se nastavují ručně na každém zařízení. Díky nedostatkům v protokolu lze relativně snadno tento klíč získat, dokonce na to existují i specializované programy.

- Šifrování pomocí WPA používá kvůli zpětné kompatibilitě WEP klíče, které jsou bezpečným způsobem dynamicky měněny. Autentizace je řešena pomocí PSK (Pre-Shared Key) – obě strany používají stejné, pokud možno dlouhé, heslo.
- Nejnovější WPA2 používá kvalitnější šifrování AES, která je náročnější na výpočetní výkon, a proto ji není možné implementovat na starších zařízeních.

Standard	Pásmo [GHz]	Maximální rychlost [Mbit/s]
IEEE 802.11	2,4	2
IEEE 802.11a	5	54
IEEE 802.11b	2,4	11
IEEE 802.11g	2,4	54
IEEE 802.11n	2,4 nebo 5	600

Tabulka 1: Přehled standardů 802.11 [14]

2.3 IQRF

IQRF je proprietární bezdrátová technologie, vyvinutá českou firmou MICRORISC s.r.o., z Jičína. Je vhodná pro telemetrii, automatizaci domů, bytů a pracovišť (IQRF Smart House), průmysl, služby i další oblasti. Základním komunikačním prvkem je modul RF transceiveru ve formátu karty SIM. Integrovaný řídicí mikrokontrolér je vybaven operačním systémem, což spolu s dalšími unikátními patentově chráněnými vlastnostmi IQRF (včetně plné podpory komplexních sítí IQMESH) umožňuje velice snadnou tvorbu aplikací i uživatelům bez znalostí VF techniky, komunikací, procesorů, programování a téměř i elektroniky vůbec.

Do bezdrátové sítě je možný přístup i zvenčí, prostřednictvím bran (gateway), které realizují interface ke komunikačním standardům, např. USB, Ethernet, GSM, RS485 nebo ZigBee, a poskytují i další funkce (vyšší výkonnost, přídavné paměti, časové informace apod.). Do sítě může komfortně přistupovat i lidská obsluha, a to nejen přes PC nebo mobilní telefon, ale i přímo branou s dotykovým displejem.[1, 6]

Technologie IQRF používá pro vysílání zpráv následující pásma:

- **868 MHz** dostupných 62 kanálů, definováno pro Evropu,
- **916 MHz** dostupných 189 kanálů, definováno pro Ameriku.

Přenosová rychlost je 1.2 kb/s – 115 kb/s, výstupní výkon až 3,5 mW.

2.3.1 Topologie sítě

2.3.1.1 Peer-to-peer Jednoduchý a nenáročný typ sítě, určený pro přímé propojení dvou a více zařízení bez řídicího uzlu. Operační systém přidává do vysílaného paketu pouze informace o jeho délce, interpretace dat paketu je kompletně v režii uživatele. Data přijímají všechna zařízení v dosahu sítě. Počet zařízení není ničím omezen.[1]

2.3.1.2 IQMESH Topologie se skládá z jednoho řídicího uzlu (Coordinator) a koncových zařízení (Nodes). Operační systém přidává do vysílaného paketu kompletní síťové informace. Pakety jsou směrovány pouze určeným stanicím. Veškerá síťová komunikace je kódovaná.[1]

2.4 ZigBee

ZigBee je bezdrátová technologie založená na standardu IEEE 802.15.4. Specifikace byla vyvinuta v roce 2004. Zjistilo se, že technologie Bluetooth není vhodná pro některé druhy aplikace, např. v průmyslu. Z tohoto důvodu vznikla ZigBee aliance, která měla vytvořit nový standard, který by byl použitelný pro účely průmyslové automatizace. ZigBee je navržen jako jednoduchá a flexibilní technologie nenáročná na spotřebu elektrické energie.[16, 13]

Hlavní využití technologie ZigBee:

- automatizace budov (zabezpečení, ovládání světel, kontrola přístupu),
- spotřební elektronika (dálkové ovládání domácí techniky),
- zabezpečovací systémy (chipové karty).

Standard ZigBee definuje následující typy zařízení:

- FFD (Full Functional Device) – u tohoto typu zařízení je implementována kompletní protokolová sada, dále pak tato zařízení poskytují všechny služby stanovené standardem.
- RFD (Reduced Functionality Device) – zařízení implementují pouze nezbytné části protokolu, aby se na minimum snížily požadavky na systémové zdroje daného zařízení. Zařízení mohou pracovat pouze jako koncové části sítě. Zařízení typu RFD mohou komunikovat pouze s koordinátorem sítě.

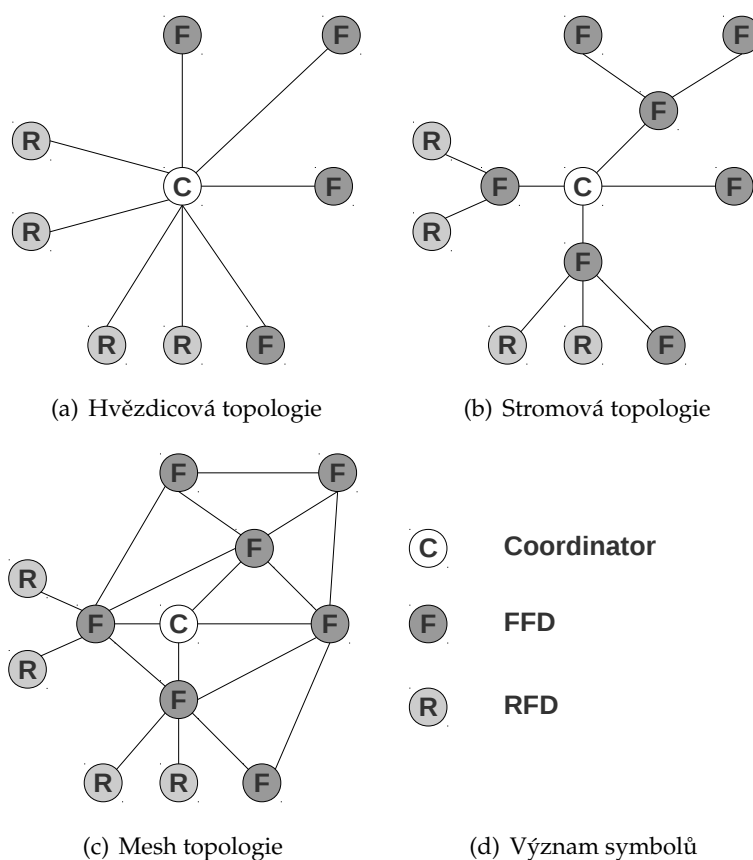
2.4.1 Topologie sítě

2.4.1.1 Hvězdicová Topologie se skládá z jednoho řídicího uzlu (PAN Coordinator) a koncových zařízení (Nodes).

2.4.1.2 Stromová Umožňuje prodloužit fyzický dosah sítě pomocí předávání zpráv mezi jednotlivými zařízeními (pouze zařízení typu FFD).

2.4.1.3 Mesh Topologie typu Mesh je kombinací předchozích uvedených typů sítí. Výhoda je možnost směřovat data jinou cestou v případě výpadku nějakého zařízení po cestě.

Každé zařízení má přidělenou MAC adresu o délce 64 bitů. Standard definuje ještě krátkou verzi o délce 16 bitů. Při použití zkráceného adresování může být v jedné síti



Obrázek 1: ZigBee topologie

maximálně 65 535 zařízení. Každá síť používá pro identifikaci 16 bitové PAN ID. Každou jednotlivou síť zakládá a následně spravuje hlavní jednotka – síťový koordinátor. Ostatní zařízení v síti pracují jako směrovače dat a koncová zařízení.[16]

2.4.2 OSI model

Specifikace 802.15.4 definuje pouze MAC a fyzickou vrstvu modelu OSI. O vyšší vrstvy (síťová, transportní) se stará ZigBee aliance. Aplikační vrstvu si definuje sám zákazník svými požadavky.

2.4.2.1 PHY (Fyzická vrstva) Bezdrátová technologie, používá následující pásma:

- pásmo ISM 2.4 GHz, 16 kanálů, přenosová rychlost 250kb/s, platí celosvětově,
- pásmo 915 MHz, 10 kanálů, přenosová rychlost 40kb/s, pouze pro americký kontinent,
- pásmo 868 MHz, 1 kanál, přenosová rychlost 20kb/s, pouze pro Evropu.

2.4.2.2 MAC (linková vrstva) Řeší, jak má vypadat komunikace mezi jednotlivými zařízeními v síti, tzv. uzly sítě. Komunikace zde probíhá pomocí čtyř typů rámců, které slouží pro přenos užitečné informace, případně pro chod sítě (sestavování spojení, správa, řízení). Na základě časové synchronizace mezi centrální stanicí a koncovou stanicí dochází u uspané koncové stanice k probouzení ve vymezeném časovém intervalu, poté jsou přeneseny veškeré potřebné informace. Interval synchronizačních sekvencí může být nastaven v rozmezí 15 ms až přibližně 15 minut.[13]

- Data Frame – rámeček pro přenos užitečné informace pro všechny datové přenosy.
- Acknowledgement Frame – rámeček pro přenos potvrzování při potvrzované komunikaci.
- MAC Command Frame – rámeček k centralizovanému konfigurování, nastavení a řízení klientských zařízení v síti.
- Beacon Frame – rámeček k synchronizaci zařízení v síti, využívá se v síťovém módu, v němž umožňuje přejít klientskému zařízení do spánkového režimu (malá spotřeba energie).

2.4.2.3 NWK (Síťová vrstva) Síťová vrstva provádí připojování k síti a odpojování od ní, zabezpečení a směrování paketů. Jako základní zabezpečení mechanismu se používá 64bitový nebo 128bitový kryptografický algoritmus AES (Advanced Encryption Standard). Dále zajišťuje nalezení zařízení v rámci jednoho přeskočení. V případě koordinátora sítě je odpovědná za start sítě a přiřazování adres nově začleněným zařízením.[13]

2.4.2.4 APL (Aplikační vrstva) Aplikační vrstva protokolu ZigBee se skládá z pomocné aplikační podvrstvy (APS), objektů ZigBee (ZDO) a uživatelských aplikačních objektů. Aplikační pomocná podvrstva umožňuje párování zařízení podle poskytovaných služeb a požadavků. Objekt ZigBee definuje roli zařízení v rámci sítě (koordinátor, směrovač nebo koncové zařízení) a spravuje poskytované služby.[16, 13]

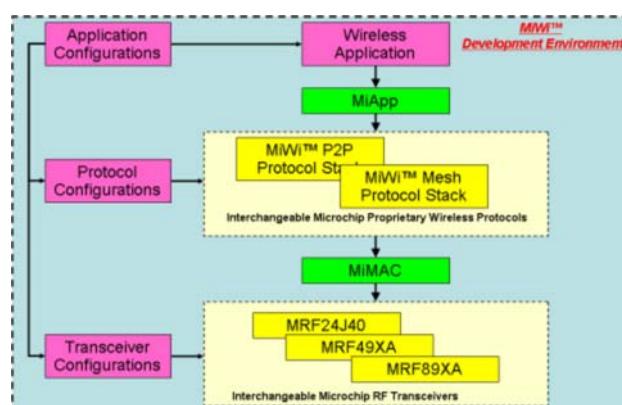
2.5 MiWi(TM)

MiWi(TM) je proprietární bezdrátová technologie firmy Microchip, která by měla pomoci zákazníkům při vývoji a rychlejším uvedení vyvíjených zařízení na trh. Pro přenos informací používá ISM pásmo. Součástí MAL (Microchip Applications Library) jsou definice MiWi Stack dostupné pro širokou škálu mikrořadičů PIC od 8-bitových, přes 16-bitovou řadu, až po 32-bitové verze. Podobně jako ZigBee využívá standardu IEEE 802.15.4. Pro přenos informací je použito ISM pásmo.[5]

2.5.1 Topologie sítě

- Hvězdicová – složena z jednoho řídicího uzlu (PAN Coordinator) a koncových zařízení (Nodes).

- **Stromová** – umožňuje prodloužit fyzický dosah sítě pomocí předávání zpráv mezi jednotlivými zařízeními (pouze zařízení typu FFD).
- **Mesh** – je kombinací předchozích uvedených typů topologií. Výhoda je možnost směřovat data jinou cestou v případě výpadku nějakého zařízení po cestě.[5]



Obrázek 2: MiWi(TM) vývojové prostředí

MiWi(TM) vývojové prostředí se skládá ze dvou vrstev[9]:

- **MiApp** – díky použití MiApp interface je možné lehce měnit typ protokolu (MiWi Mesh nebo MiWi P2P) beze změny firmware.
- Použitím **MiMAC** vrstvy je možné lehce měnit typ použitého RF modulu.

2.6 Srovnání technologií

Standard	Bluetooth	WiFi	ZigBee a MiWi	IQRF
Aplikační zaměření	náhrada za kabel	připojení internetu	automatizace	automatizace
Systémové zdroje (ROM)	250KB a více	1MB a více	4 KB –32KB	–
Životnost baterie (dnů)	1 – 7	0,5 – 5	100 – 1 000 i více	–
Max. velikost sítě	7	32	65 000	65 000
Přenosová rychlost (kb/s)	720	11 000	20 –250	1,2 – 115

Tabulka 2: Porovnání bezdrátových technologií dle parametrů

2.6.1 WiFi

Nevýhoda technologie WiFi je především v náročnosti na systémové zdroje a relativně vysoká spotřeba. Výhoda je vysoká přenosová rychlost a snadná dostupnost.

2.6.2 Bluetooth

Nevýhoda Bluetooth je možnost spojit navzájem pouze osm zařízení (v ideálním případě, v reálných podmínkách bývá problém propojit i méně zařízení). Výhoda Bluetooth – je implementováno u většiny současných mobilních zařízení.

2.6.3 IQRf

Podle mého názoru je nevýhoda IQRf v tom, že jej vyvíjí pouze jediná firma, nejedná se o standard a firma může kdykoliv z jakéhokoliv důvodu ukončit podporu. Výhoda je snadná implementace zákaznické aplikace – vše řeší operační systém, zákazník řeší pouze to, na co má aplikace reagovat, a od samotné implementace je odstíněn.

2.6.4 MiWi

Nevýhoda řešení – technologie je proprietární a podporuje pouze firma Microchip. Výhoda snadná implementace, použití stejných, modulů jako používá ZigBee.

2.6.5 ZigBee

Nevýhoda je, že ZigBee je technologie relativně nová, ještě nedoladěná, zatím není moc rozšířená. Výhoda je integrování zabezpečení metodou AES, nízká cena modulů, dobře známá dokumentace.

3 Technologie iDům

3.1 Popis systému iDům

iDům je systém inteligentního domu, který je vyvíjen na katedře telekomunikační techniky Vysoké školy Báňské ve spolupráci s firmou ELEKTRO-FA. PAVELEK, s.r.o. s ohledem na maximální funčnost, jednoduchost a hlavně co nejnižší cenu oproti jiným řešením.

Systém může pracovat ve dvou módech:

- systém bez řízení,
- systém s centrálním řízením.

3.1.1 Systém bez řízení

Je levnější a jednodušší variantou, vhodný např. pro domovní zvonky, řízení osvětlení a jednoduchá zabezpečovací zařízení. Každé zařízení má naprogramovanou adresu, případně seznam adres, na které má posílat své informace, nebo je adresa volena akcí, například zadáním čísla na klávesnici domovního tabla. Zprávy se předávají konkrétním zařízením.

Konfiguraci zařízení v tomto případě musí provádět kvalifikovaná osoba a je k tomu třeba hardwarové vybavení v podobě notebooku a modulu iDům konfigurátor. Z toho vyplývá, že po naprogramování systému již uživatel nemá žádnou možnost jakkoliv systém upravovat, vždy je třeba pro každou uživatelskou změnu kvalifikovaná osoba.[4]

3.1.2 Systém s centrálním řízením

Jádrem systému s řízením je miniaturní počítač s nízkou spotřebou, na kterém běží upravená verze operačního systému linux a uživatel tak má možnost nastavovat všechny parametry systému iDům, případně změnit celou jeho funkci. K počítači lze připojit např. LCD display, touchpanel, případně jiné periférie. Konfiguraci lze provádět i na dálku např. přes internet, takto je také možno řídit spotřebiče v domácnosti.[4]

Systém iDům se skládá z několika bloků:

- vstupní prvky (vypínače, teploměry, čidla světla, čidlo povětrnostních podmínek),
- akční prvky (relé, řízení stmívačů, ovládání brány),
- zabezpečovací prvky, fungující nezávisle na řídícím PC (PIR, dveřní kontakty, siréna),
- prvky domácího telefonu, omezeně fungující bez funkčního řídícího PC (tablo, interkom),
- řídící PC.

3.2 Komunikace

Hardwarově je sběrnice řešena co nejjednodušeji, jedná se o čtyřvodičovou sběrnici, která obsahuje[4]:

- dva napájecí vodiče (+) a (–),
- datový vodič (D) pro přenos signalizace,
- audio vodič (A) pro přenos specifické informace (audio, video).

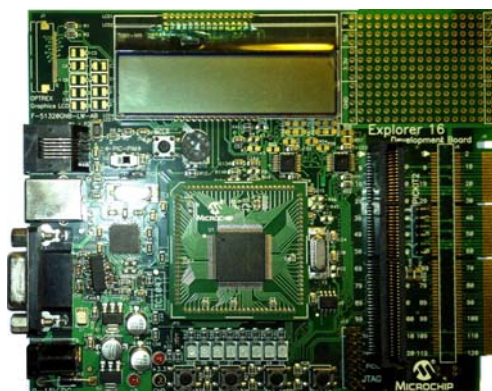
Komunikuje se pomocí jednoduchého bezstavového protokolu, data jsou vysílána vždy po 8 bitech stálou rychlostí 19,2 kbps. Zprávy systému iDům mají proměnnou délku. Jako první se vysílá hlavička zprávy a poté tělo. Hlavička je vždy stejná. Obsahuje údaje o typu zprávy, identifikaci cílového (přijímacího) a zdrojového (vysílacího) zařízení, délku zprávy (počet všech vyslaných bytů včetně hlavičky) a uživatelskou informaci. Tělo zprávy je proměnné délky. Na konci zprávy je kontrolní součet. Každá zpráva musí být potvrzena odpovědí.[4]

Typ zprávy	Cílové zařízení	Zdrojové zařízení	Délka zprávy	Uživatelský byte	Tělo	CRC
------------	-----------------	-------------------	--------------	------------------	------	-----

Tabulka 3: Formát iDům zprávy [4]

4 Vývoj

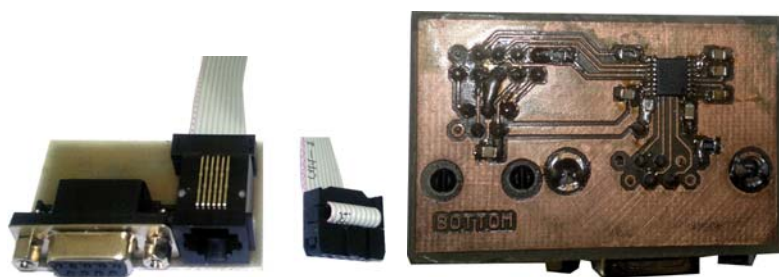
Vývoj bezdrátových modulů probíhal na zapůjčených vývojových kitech Explorer 16 Development Board¹ firmy Microchip. Osazen byl referenční mikrořadič PIC24FJ128-GA010.[11] Explorer 16 je vybaven rozšiřujícím portem PICTail(TM) Plus, do kterého se umísťuje karta se ZigBee modulem MRF24J40MA.²



Obrázek 3: Vývojový kit explorer16

4.1 Programování a ladění modulů

Aby bylo možné moduly naprogramovat a ladit na počítači, vyrobil jsem si k tomuto účelu propojovací modul. Propojovací modul se připojuje k bezdrátovým modulům pomocí desetipinového dvouřadého konektoru. Schéma zapojení je na obrázku 23.



(a) Pohled na stranu součástek

(b) Pohled na stranu plošného spoje

Obrázek 4: Propojovací modul

¹DM240001 <http://www.microchipdirect.com/ProductSearch.aspx?Keywords=DM240001>

²AC164134 <http://www.microchipdirect.com/ProductSearch.aspx?Keywords=AC164134>

4.2 Microchip ZigBee stack 2006

Je to balíček zdrojových kódů, nápovědy a aplikací, který slouží pro mikrořadiče microchip rodiny PIC18 a PIC24.³ Díky těmto zdrojovým kódům není třeba studovat celou normu 802.11.15[15] a vyvíjet aplikaci od nuly. Je třeba se seznámit s dokumentem[8], kde je popsáno, jakým způsobem se s Microchip ZigBee stack 2006 pracuje. Součástí Microchip ZigBee stack 2006 je také sada demonstračních projektů, které slouží pro ukázkovou implementaci jednotlivých typů bezdrátových zařízení. Tyto demonstrační projekty jsem využil pro inspiraci při tvorbě svého projektu. Pro překlad zdrojových kódů je použit překladač microchip MPLAB® C Compiler for PIC24 MCUs and dsPIC DSCs, zkráceně označovaný jako MPLAB C30⁴, ve verzi lite, který je oproti verzi standard bezplatný a nemá provedenou optimalizaci kódů – tzn., že výsledný kód zabírá více místa v paměti mikrořadiče. Velikou oporou je kniha [7], ve které je popsána práce s mikrořadičem PIC24-FJ128GA010 a překladačem C30 s názornými příklady krok za krokem.

Každá aplikace, která využívá Microchip ZigBee stack 2006, musí mít základní strukturu programu uvedenou ve výpise č. 1. Po spuštění aplikace je třeba jako první inicializovat ZigBee modul MRF24J40MA metodou `ZigBeeInit()`. Aplikace dále běží ve smyčce a periodicky volá funkce[8]:

- `ZigBeeTasks(¤tPrimitive)` funkce řeší zpracování ZigBee primitiv,⁵
- `ProcessZigBeePrimitives()` vyhodnotí, která následující ZigBee Primitiva je třeba zpracovat,
- `ProcessNONZigBeeTasks()` metoda zpracovává činnosti, které nesouvisí se ZigBee. U našich aplikací je to hlavně obsluha sběrnice iDům.

```
while (1)
{
    CLRWDT();

    ZigBeeTasks( &currentPrimitive );

    ProcessZigBeePrimitives();

    ProcessNONZigBeeTasks();
}
```

Výpis 1: Základní struktura programu

³Instalační balíček lze stáhnout ze stránek http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en538049

⁴Překladač je možné stáhnout z http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en535364

⁵Seznam všech primitiv, které funkce zpracovává je uveden v [15]

4.2.1 Příjem ZigBee zprávy

ZigBee stack informuje aplikaci o přijaté zprávě pomocí primitivy `APSDE_DATA.indication`. Příjem zprávy se uskutečňuje ve funkci `ProcessZigBeePrimitives(void)`. Zpráva je rozdělena po jednom bajtu a načtení bajtu z bufferu se provádí pomocí funkce `APLGet()`. Jakmile je přijatá zpráva přečtena z bufferu a zpracována, je třeba přijatá data zlikvidovat pomocí funkce `APLDiscard()`, pokud by se tak nestalo, aplikace by nemohla zpracovat další příchozí zprávu. Příjem zprávy je zobrazen v ukázce zdrojového kódu č. 2 na straně 18.

```

case APSDE_DATA.indication:
{
    // Declare variables used by this primitive .
    currentPrimitive = NO_PRIMITIVE; // This may change during processing.
    frameHeader = APLGet();
    switch (params.APSDE_DATA.indication.DstEndpoint)
    {
        case EP_ZDO:
            // Handle all ZDO responses to requests we sent.
            break;
            // Include cases for all application endpoints.
        }
        APLDiscard();
    }
    break;

```

Výpis 2: Ukázka příjmu ZigBee zprávy

4.2.2 Odeslání ZigBee zprávy

Při odesílání zprávy musíme splnit několik kroků:

- zkontrolovat, zda je aplikační vrstva schopna odeslat zprávu, `ZigBeeReady()` musí vrátit `TRUE`,
- zablokovat systém pomocí `ZigBeeBlockTx()`, zavolání `ZigBeeReady()` by mělo vrátit `FALSE`,
- načíst zprávu do pole `TxBuffer`, `TxData` je použit jako index k procházení pole. Jakmile je zpráva načtena, `TxData` musí ukazovat na první pozici za zprávou,
- pro správné zpracování dat na příjmací straně je důležité nastavit všechny potřebné parametry primitivy `APSDE_DATA.request`,
- nastavit `currentPrimitive` na `APSDE_DATA.request` a zavolat `ZigBeeTasks()`.

Informace o stavu odeslání zprávy je vrácena pomocí `APSDE_DATA.confirm`. Pokud se zprávu nepodaří odeslat, ZigBee stack zkusí automaticky opakovat odeslání zprávy.[8] Maximální počet pokusů je roven `apsMaxFrameRetries`. Odeslání zprávy je zobrazeno v ukázce č. 3 na straně 19.

```

if (ZigBeeReady())
{
    /* Send a message to device whose network address is destinationAddress */
    if ( myStatusFlags.bits.bmyBottonWasPressed )
    {
        myStatusFlags.bits.bmyBottonWasPressed = FALSE;
        /* Block the transmit path before loading the message payload */
        ZigBeeBlockTx();
        /* requesting 10-bytes of data from the receiving device */
        TxBuffer[TxData++] = 0x0a;
        /* Use unicast, 16-bit network address */
        params.APSDE_DATA.request.DstAddrMode = APS_ADDRESS_16_BIT;
        /* Destination address of devices of interest, including endpoint */
        params.APSDE_DATA.request.DstAddress.ShortAddr = destinationAddress;
        params.APSDE_DATA.request.DstEndPoint = destinationEndPoint;
        /* Using the ZCP profileID and ClusterIDs as an example */
        params.APSDE_DATA.request.ProfileId.Val = MY_PROFILE_ID;
        params.APSDE_DATA.request.ClusterId.Val = BUFFER_TEST_REQUEST_CLUSTER;

        params.APSDE_DATA.request.SrcEndpoint = EP_BUTTON_APPLET;

        params.APSDE_DATA.request.RadiusCounter = DEFAULT_RADIUS;

        /* Find a route if possible */
        params.APSDE_DATA.request.DiscoverRoute = ROUTE_DISCOVERY_ENABLE;
        params.APSDE_DATA.request.TxOptions.Val = 0;
        /* request an ACK from application on receiving device */
        params.APSDE_DATA.request.TxOptions.bits.acknowledged = 1;

        /* Trigger the next primitive to be executed */
        currentPrimitive = APSDE_DATA.request;
    }
}

```

Výpis 3: Ukázka odeslání ZigBee zprávy

4.2.3 Zabezpečení přenosu informací

Microchip ZigBee stack 2006 podporuje všech sedm typů zabezpečení definovaných v ZigBee specifikaci.[8]

Zabezpečení se dá rozčlenit do následujících tří typů:

- Message Integrity Code (MIC) – režim kontroly integrity paketu. MIC část se přidává k odesílanému paketu, což zajistí, že celý paket včetně hlavičky a samotné zprávy nebyl jakkoliv změněn během přenosu. V tomto režimu není zpráva šifrována.
- Encryption (ENC) – režim zabezpečení zprávy formou šifrování. Obsah zprávy nelze získat bez šifrovacího klíče. Tento režim nedokáže zkontrolovat integritu rámce, obsah hlavičky, ani zdroj, odkud byl paket odeslán.

- ENC-MIC – režim je kombinací pomocí předchozích dvou typů. V tomto režimu je zpráva šifrována. Integrita hlavičky i zprávy je chráněna MIC připojeným na konec paketu.

Dále existuje režim bez zabezpečení (0x00) – v tomto režimu je část starající se o zabezpečení vypnuta.

V tabulce č. 4 jsou uvedeny podporované bezpečnostní režimy. Pro odeslání zabezpečené zprávy je třeba doplnit do kódu pro odeslání zprávy tento řádek `params.APSDE_DATA_request.TxOptions.bits.securityEnabled = 1;`. Odeslání zabezpečené zprávy je zobrazeno v ukázce č. 4 na straně 20.

```

if (ZigBeeReady() )
{
    if (bLightSwitchToggled)
    {
        bLightSwitchToggled = FALSE;
        ZigBeeBlockTx();
        TxBuffer[TxData++] = 0x08
        /*request eight bytes*/

        params.APSDE_DATA_request.DstAddrMode = APS_ADDRESS_16_BIT;
        params.APSDE_DATA_request.DstEndpoint = destinationEndpoint;
        params.APSDE_DATA_request.DstAddress.ShortAddr = destinationAddress;
        params.APSDE_DATA_request.ProfileId.Val = MY_PROFILE_ID;
        params.APSDE_DATA_request.RadiusCounter = DEFAULT_RADIUS;
        params.APSDE_DATA_request.DiscoverRoute = ROUTE_DISCOVERY_ENABLE;
        params.APSDE_DATA_request.TxOptions.Val = 0;

        /*zapnutí zabezpečení při odeslání zprávy*/
        params.APSDE_DATA_request.TxOptions.bits.securityEnabled = 1;

        params.APSDE_DATA_request.SrcEndpoint = EP_SWITCH;
        params.APSDE_DATA_request.ClusterId.Val = BUFFER_TEST_REQUEST_CLUSTER;
        currentPrimitive = APSDE_DATA_request;
    }
}

```

Výpis 4: Ukázka odeslání zabezpečené ZigBee zprávy

4.3 Konfigurace bezdrátových zařízení

Součástí Microchip ZigBee stack® je aplikace Microchip ZENA(TM). Tato aplikace slouží pro konfiguraci každého bezdrátového zařízení, které používá Microchip ZigBee® stack. Dále je aplikace užitečná pro zachytávání a analýzu síťové komunikace. Pro analýzu je třeba dokoupit potřebný hardware.⁶

Po spuštění aplikace ZENA.exe⁷ vybereme v menu položku ZigBee(TM) Tools – Stack Configuration. Otevře se nám nové okno, které je rozděleno na osm záložek, každá zálož-

⁶DM183023 - ZENA(TM) Network Analyzer <http://www.microchipdirect.com/productsearch.aspx?Keywords=DM183023>

⁷Při výchozí instalaci je umístěna v C:\Microchip\Solutions\ZigBee2006Res\ZENA.exe

Bezpečnostní mód		Typ zabezpečení			délka MIC
Identifikátor	Jméno	Kontrola přístupu	šifrování dat	integrita rámce	
0x01	MIC-32	X		X	4
0x02	MIC-64	X		X	8
0x03	MIC-128	X		X	16
0x04	ENC	X	X		0
0x05	ENC-MIC-32	X	X	X	4
0x06	ENC-MIC-64	X	X	X	8
0x06	ENC-MIC-128	X	X	X	16

Tabulka 4: Porovnání bezpečnostních režimů[8]

ka se zabývá určitou oblastí konfigurace bezdrátového zařízení. Na obrázku 5 na straně 23 je ukázáno rozhraní aplikace ZENA.

4.3.1 Záložka ZigBee Device

- **MAC Address** – 64bitová MAC adresa, která se skládá z unikátního OUI (24-bitů) a čísla (40-bitů), které si výrobce určuje dle svých vnitřních norem. Toto rozdělení je stejné jako u jiných technologií (bluetooth, ethernet) a mělo by zabránit, aby zařízení od různých výrobců mělo stejnou adresu. Aby se naše vyvíjené zařízení dalo použít pro komerční účely, bylo by třeba nechat si přidělit prefix od organizace IEEE.⁸ Pokud již nějaká organizace má OUI identifikátor přidělen (např. pro ethernet), není třeba se znovu registrovat. Pro vývoj použijeme OUI přiděleného firmě Microchip Technology, Inc. (00-04-A3(hex)).
- **ZigBee Device Type** – slouží pro výběr typu zařízení. Funkce **ZigBee Coordinator** je nastavena pro modul rozhraní iDům – ZigBee. **ZigBee End Device** je nastaveno u všech ostatních modulů (vypínač, panel s LCD atd.).
- **IEEE Device Type** – význam položky je vysvětlen v kapitole 2.4, **FFD** nastaven u modulu rozhraní iDům – ZigBee, **RFD** nastaven na všech ostatních modulech.
- **Transceiver Power** – **always On** u modulu rozhraní iDům – ZigBee, u ostatních modulů nastaveno **On When Stimulated** (modul vypínač) nebo **Periodically On** (modul LCD).
- **Available Power Sources** – u modulu rozhraní iDům – ZigBee (je napájen pomocí iDům sběrnice) nastavíme **Mains Power**, u ostatních modulů nastavíme **Disposable Battery**.

⁸<http://standards.ieee.org/develop/regauth/oui/index.html>

4.3.2 Záložka Transciever

- **Transciever** – určuje, jaký RF modul bude použit, v našem případě používáme Microchip MRF24J40.
- **Output Power** – nastavení vysílacího výkonu můžeme ponechat 0 dBm. Ovlivňuje maximální dosah signálu.
- **Pin Assignments** – nastavení mapování pinů ZigBee modulu na zvolený typ mikrořadiče.
- **Allowed Channels** – povolené kanály, na kterých může zařízení vysílat/přijímat data. V našem případě je nastaven pouze jeden libovolný kanál, čím více kanálů je vybráno, tím déle trvá inicializace bezdrátové sítě.

4.3.3 Záložka Endpoints

Definuje logické propojení přenášených dat mezi jednotlivými zařízeními, používáme profil zTest.h.

4.3.4 Záložka Security

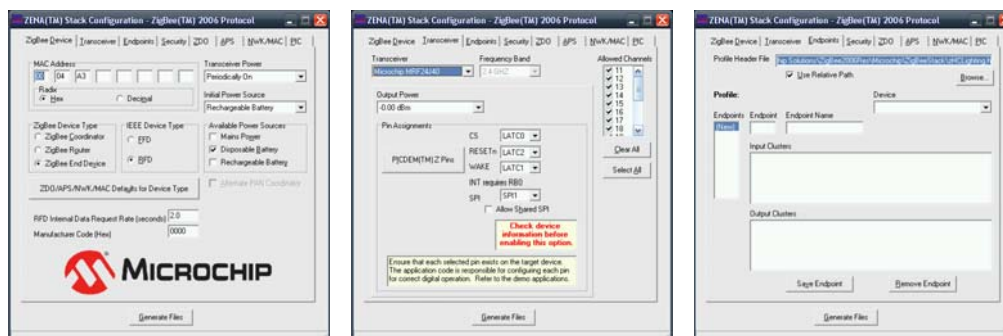
Slouží pro nastavení šifrování přenášených dat.⁹ V našem případě je zvolen **security mode** typu **residential**.

- **Trust Center** – u modulu rozhraní iDům – ZigBee je zvolena volba **I am a Trust center**. U všech ostatních modulů se musí nastavit MAC adresa modulu rozhraní iDům – ZigBee.
- **Network Key** – „tajný“ klíč, který šifruje komunikaci.

4.3.5 Záložka PIC

Slouží pro nastavení informací o použitém mikrořadiči.

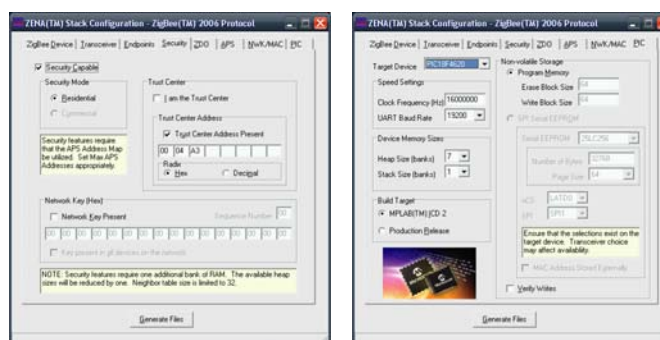
- **Target Device** – použitý typ mikrořadiče, nastavíme PIC24/dsPIC.
- **Clock Frequency (Hz)** – frekvence krystalu, nastavíme 8 000 000 Hz.
- **UART Baud Rate** – nastavení rychlosti sériové sběrnice pro textovou komunikaci s počítačem, 19 200 kbps.
- **Device Memory Size** – nastavena výchozí hodnota 2048.
- **Non-volatile Storage** – mikrořadič PIC24FJ nemá vnitřní paměť, která by zůstala zachována po vypnutí, resp. má, ale počet přepisů je dle datasheetu[12] omezen na 10 000, proto je třeba připojit externí paměť typu EEPROM přes sběrnici SPI. případně je použita paměť 25LC256.[17]



(a) Zálôžka ZigBee Device

(b) Zálôžka Transceiver

(c) Zálôžka Endpoints



(d) Zálôžka Security

(e) Zálôžka PIC

Obrázek 5: Aplikace ZENA a její zálôžky

Pokud jsme vyplnili všechny položky, můžeme kliknout na tlačítko Generate Files. Program vygeneruje soubory s konfigurací, které vložíme do našeho projektu. Jedná se o soubory myZigBee.c a ZigBee.def – v příloze č. 9 a 10 je ukázka konfiguračních souborů.

⁹Více informací o zabezpečení viz. [8] sekce Secure Transmission na straně 23

5 Vyvíjené moduly

5.1 Vlastnosti společné všem modulům

Každý modul je vybaven desetipinovým dvouřadým konektorem, který sdružuje sériový port a rozhraní ICSP.

pin	Rozhraní	funkce	popis
1	ICSP	MCLR	Master Clear (device Reset) input
3		Vcc	napájení (3,3V nebo 5V)
5		GND	zem (0V)
7		PGD	Data
9		PGC	Clock
2	UART	RX	Receive
4		TX	Transmit
6		RTS	Ready to send
8		CTS	Clear to send
10		NC	Nezapojeno

Tabulka 5: Význam jednotlivých pinů na desetipinovém dvouřadém konektoru

5.1.1 Sériový port

Umožňuje textovou komunikaci modulu s počítačem – vhodné zejména při ladění, dále pak pro nastavování parametrů a výpis stavových informací. Pro připojení k počítači je třeba mít tzv. null modem kabel[2] s podporou Full handshakingu. Dále je třeba použít převodník napěťových úrovní sériové linky typu MAX3232, který převádí napěťové úrovně mikrořadiče (0 V – 3 V) na standardní úroveň (–12 V – +12 V) používanou v počítači. V počítači je třeba spustit vhodný program.¹⁰ V programu je třeba nastavit parametry sériové linky, přenosová rychlost viz. 4.3.5, počet datových bitů 8, počet stop bitů 1, parita žádná. Pokud je vše v pořádku, měli bychom vidět okno programu, podobné tomu na obrázku 6.

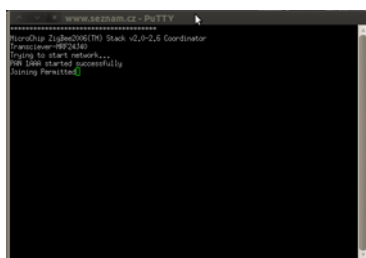
5.1.2 ICSP

Pomocí tohoto rozhraní je možné přenést přeložený program z počítače do mikrořadiče. Další funkcí rozhraní je debugging – ladění programu.

5.1.3 Paměť EEPROM

Každý modul používá paměť EEPROM firmy Microchip typu 25LC256. Externí paměť byla zvolena záměrně, aby se relativně častými zápisy neopotřebovávala paměť mikro-

¹⁰Např. PuTTY, který lze stáhnout z <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>



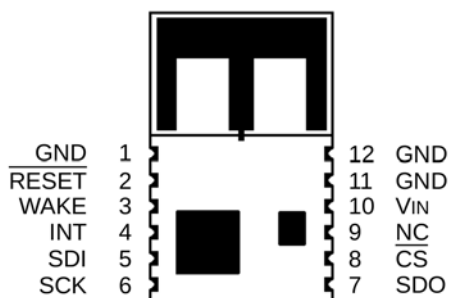
Obrázek 6: Komunikace s modulem pomocí sériového portu

řadiče. Počet přepisů paměti EEPROM by měl být dle datasheetu[17] roven jednomu miliónu, n rozdíl od FLASH paměti mikrořadiče[12], kde je počet zápisů roven 10 000.

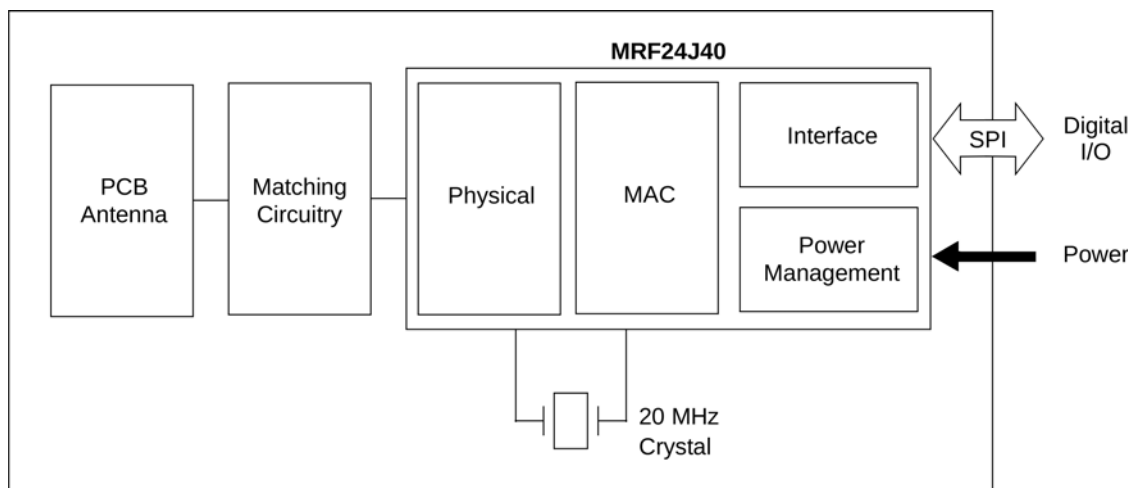
5.1.4 RF modul



(a) Fotografie modulu



(b) Zapojení vývodů



(c) Blokový diagram

Obrázek 7: RF modul MFR24J40MA

Každý modul je vybaven RF transceiverem MRF24FJ40MA od firmy Microchip. Vlastnosti modulu:

- podpora pro ZigBee(r), MiWi(TM), MiWi P2P,
- komunikace pomocí SPI sběrnice,
- nízká spotřeba:
 - RX režim 19 mA,
 - TX režim 23 mA,
 - sleep režim 2 μ A,
- napájecí napětí 2,4 – 3.6 V (typicky 3.3 V),
- přenosová rychlost 250 kbps.

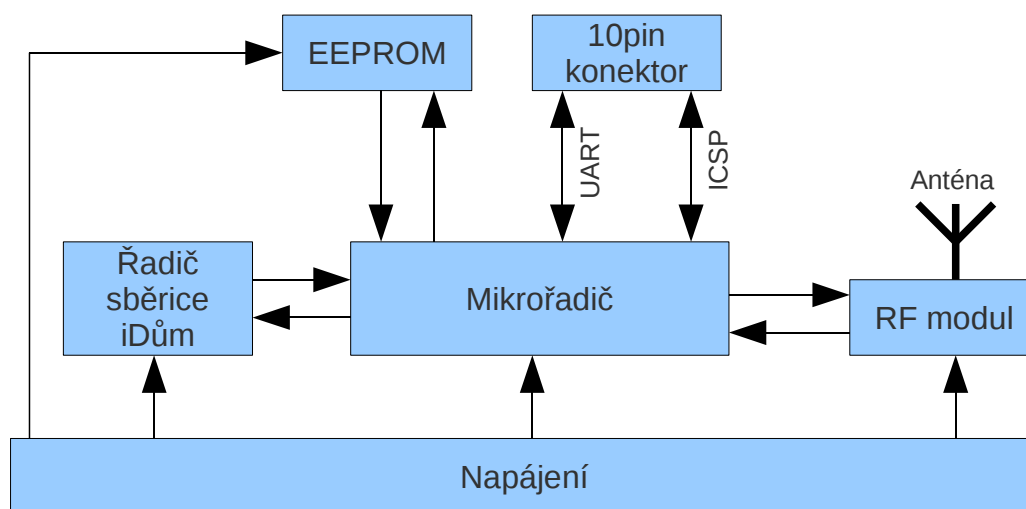
Více informací o modulu je uvedeno v jeho dokumentaci.[10]

5.2 Modul rozhraní iDům - ZigBee

5.2.1 Popis

Modul rozhraní iDům - ZigBee je v systému pouze jeden a má za úkol zejména:

- sestavit ZigBee síť,
- přijímat a zpracovávat požadavky připojených zařízení,
- překládat zprávy systému iDům a přeposílat je jednotlivým bezdrátovým zařízením.



Obrázek 8: Blokový diagram modulu rozhraní iDům – ZigBee

Velikost plošného spoje je navržena tak, aby se modul dal umístit do elektromontážní krabičky KU68. Na obrázku 10 je zobrazen plošný spoj modulu, na obrázku 11 je vyfocen vyrobený prototyp, na obrázku 20 je schéma.

5.2.2 Předávání zpráv

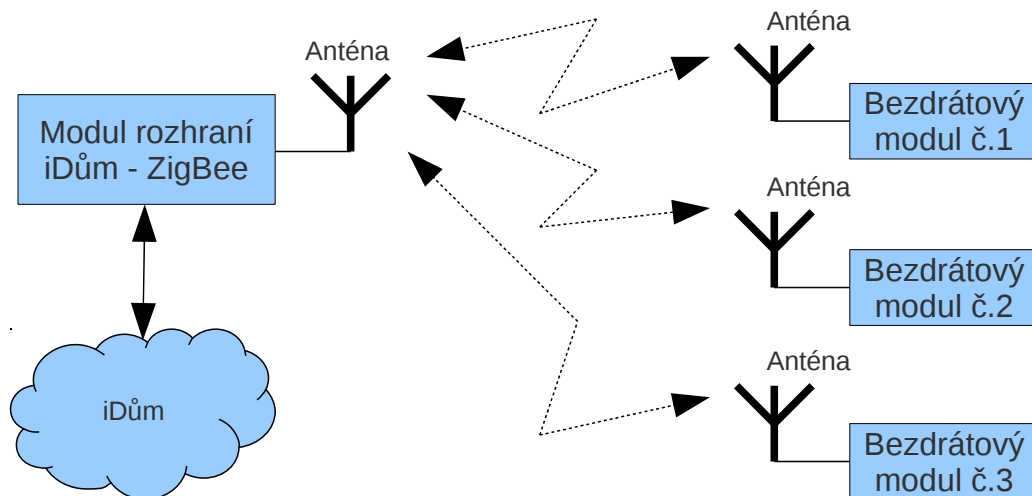
Předávání zpráv je řešeno pomocí struktury `wirelessDevInfo`¹¹, definované v souboru `iDumProfile.h`, kde jsou uloženy stavové informace a jednotlivé zprávy, které se mají odeslat ze systému iDům do bezdrátové části, a zprávy, které se odesílají z bezdrátové části do systému iDům. Na obrázku 18 strana 43 je znázorněno zpracování příchozí zprávy ze systému iDům.

```

// Store asociated wireless devices
struct
{
    BYTE    nrOfDevices;

```

¹¹Viz. výpis 5 na straně 27



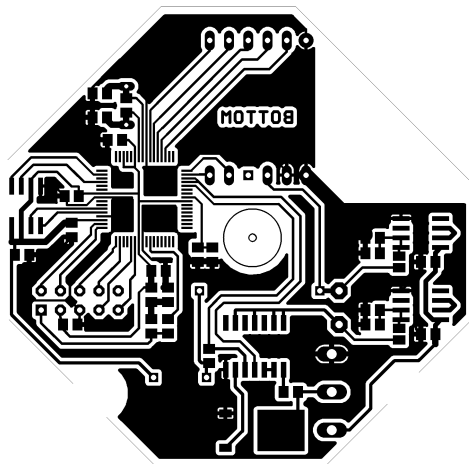
Obrázek 9: Znázornění funkce modulu rozhraní iDům – ZigBee

```

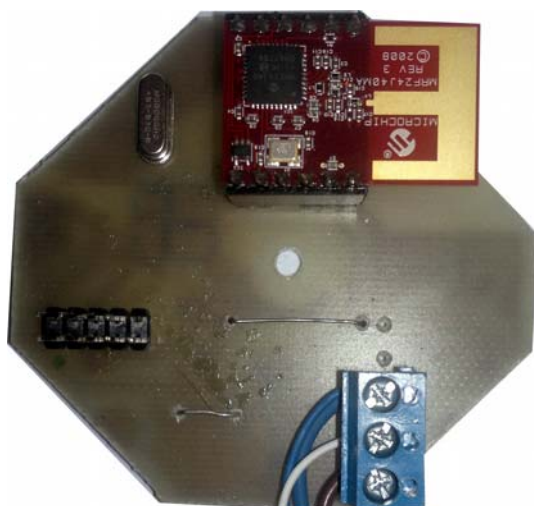
struct
{
    BYTE devReady;    //Flag 1 pokud je zarizeni inicializovane (ma prideleny adresy)
    BYTE devUsed;    //pokud je zarizeni pouzite aby se neprepsalo 1
    BYTE devIDRequest; //1 = pozadavek pro ziskani devID
    BYTE zpravaiPrijata; //Byla prijata zprava z idum
    BYTE zpravawPrijata; //Byla prijata zprava ze zigbee
}devState [MAX_NEIGHBORS];
struct
{
    SHORT_ADDR wDevADR[MAX_NEIGHBORS]; //adresa zigbee
    BYTE iDevADR[MAX_NEIGHBORS]; //adresa idum
} adr;
struct
{
    TZprava zprava;
    TZprava vysilani;
}data[MAX_NEIGHBORS];
} wirelessDevInfo;

```

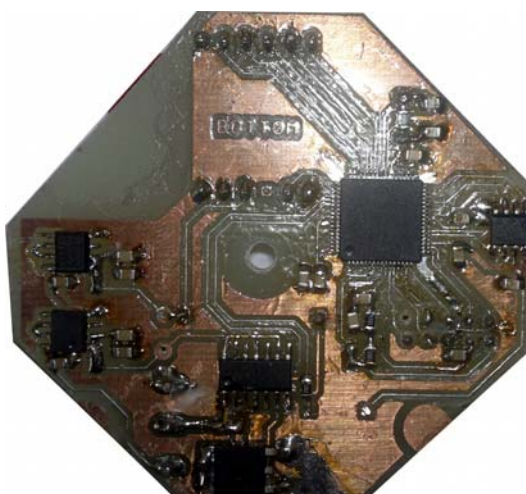
Výpis 5: Struktura wirelessDevInfo



Obrázek 10: Plošný spoj modulu rozhraní iDům – ZigBee



(a) Pohled na stranu součástek



(b) Pohled na stranu plošného spoje

Obrázek 11: Hotový modul rozhraní iDům – ZigBee

5.3 Modul vypínač

Modul vypínač dokáže ovládat až čtyři nezávislé okruhy. Pokud by bylo třeba ovládat více okruhů, byla by potřeba použít více vstupů u mikrořadiče a následně přepsat program.

5.3.1 Příjmané zprávy

Modul přijímá pouze konfigurační zprávy, na které reaguje.

5.3.2 Odesílané zprávy

Při každé změně stavu na vstupu odesílá zprávu typu `msgByte`, ve které je uveden stav vstupů (sepnuto/rozepnuto).

```
// Process the pushbutton interrupts
if ( (PBSW_00 == 0) || (PBSW_01 == 0) || (PBSW_02 == 0) || (PBSW_03 == 0))
{
    /* wait debounce time before taking any action again */
    if (PBSW_pressed == FALSE)
    {
        PBSW_pressed = TRUE;
        BYTE sendSWstrate = 0;
        //BYTE delka = 0;
        if (PBSW_00 == 0) {sendSWstrate += 1;}
        if (PBSW_01 == 0) {sendSWstrate += 2;}
        if (PBSW_02 == 0) {sendSWstrate += 4;}
        if (PBSW_03 == 0) {sendSWstrate += 8;}

        myiDevInfo.vysilani.Data.Typ=msgByte;
        myiDevInfo.vysilani.Data.Komu = defaultIDADR;
        myiDevInfo.vysilani.Data.Odkud = myiDevInfo.devID;
        myiDevInfo.vysilani.Data.UserID =0x00;
        myiDevInfo.vysilani.Data.Delka = 0x08;
        myiDevInfo.vysilani.Data.Obsah.ZByte.ByteID=0x00;
        myiDevInfo.vysilani.Data.Obsah.ZByte.Value = sendSWstrate;
        vysliiData ();
    }
}

void vysliiData (void)
{
    ZigBeeBlockTx();
    BYTE ii;
    printf ("odesilam_data:");
    for( ii = 0; ii < myiDevInfo.vysilani.Data.Delka; ii++)
    {
        TxBuffer[TxData++] = myiDevInfo.vysilani.QuickAccess[ii];
        PrintChar( myiDevInfo.vysilani.QuickAccess[ii] );
        printf (" ");
    }
}
```

```

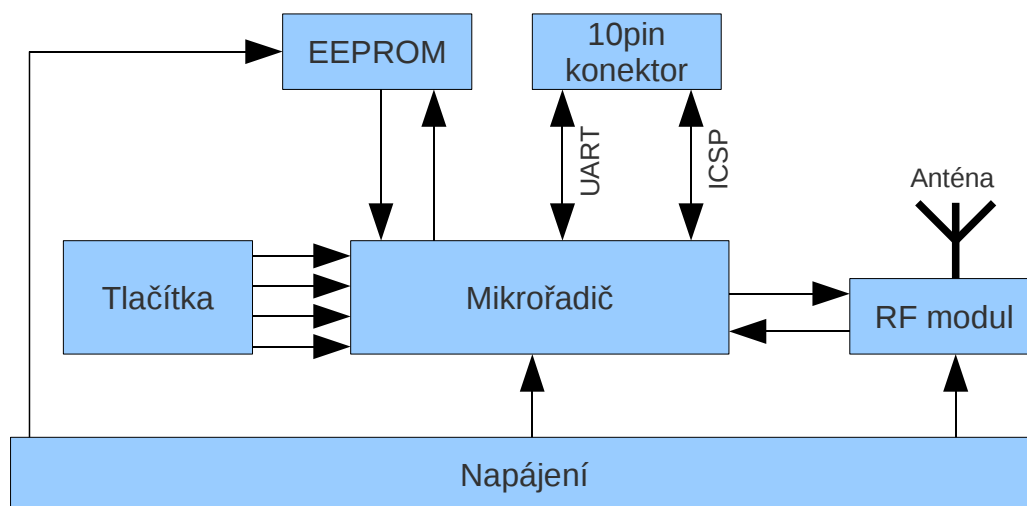
params.APSDE_DATA_request.DstAddrMode = APS_ADDRESS_16_BIT;
params.APSDE_DATA_request.DstAddress.ShortAddr = wirelessDevInfo.adr.wDevADR[i];
params.APSDE_DATA_request.RadiusCounter = DEFAULT_RADIUS;

#ifdef I_SUPPORT_SECURITY
    params.APSDE_DATA_request.TxOptions.Val = 1;
#else
    params.APSDE_DATA_request.TxOptions.Val = 0;
#endif

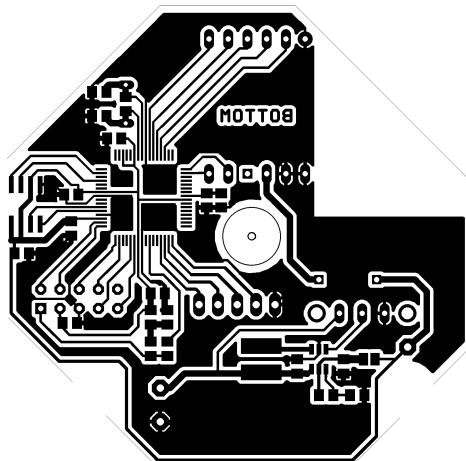
params.APSDE_DATA_request.DstEndpoint = EP_idum;
params.APSDE_DATA_request.ClusterId.Val = myiDevInfo.vysilani.Data.Typ;
currentPrimitive = APSDE_DATA_request;
printf ("\r\n");
myiDevInfo.zpravaPrijata = FALSE;
}

```

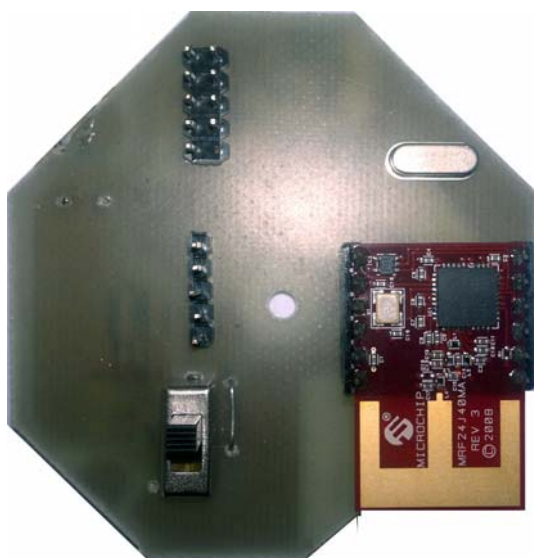
Výpis 6: Odeslání zprávy o změně stavu



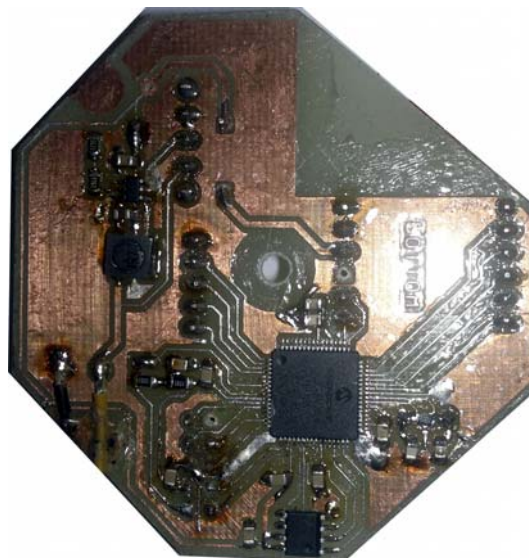
Obrázek 12: Blokový diagram modulu vypínač



Obrázek 13: Plošný spoj modulu vypínač



(a) Pohled na stranu součástek



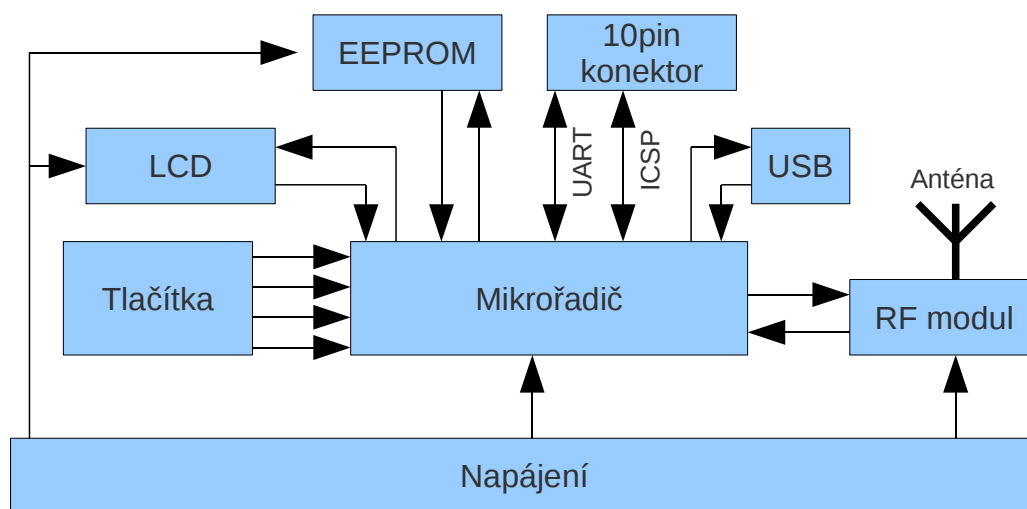
(b) Pohled na stranu plošného spoje

Obrázek 14: Hotový modul vypínač

5.4 Modul LCD

Modul LCD byl vyvinut hlavně pro demonstraci možností, je možné ho použít pro vývoj jiných typů bezdrátových modulů a jejich testování. Modul LCD je vybaven:

- textovým LCD displayem, který dokáže zobrazit 16 znaků ve dvou řádcích,
- čtyřmi tlačítky, které je možné naprogramovat, např. pohyb v textu, procházení přijatých zpráv atd.,
- USB rozhraním.



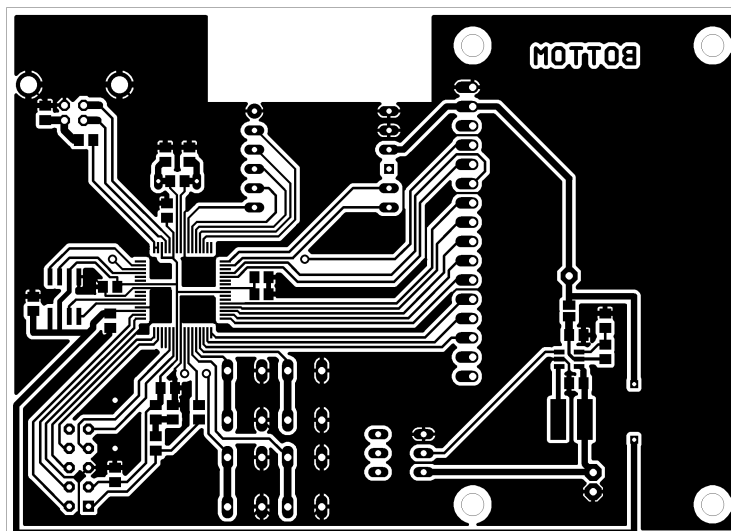
Obrázek 15: Blokový diagram modulu LCD

5.4.1 Příjmané zprávy

Modul přijímá konfigurační zprávy, na které reaguje. Dále očekává zprávu typu `msg-ByteArray`. Jakmile tento typ zprávy přijme, zobrazí ji na LCD display.

5.4.2 Odesílané zprávy

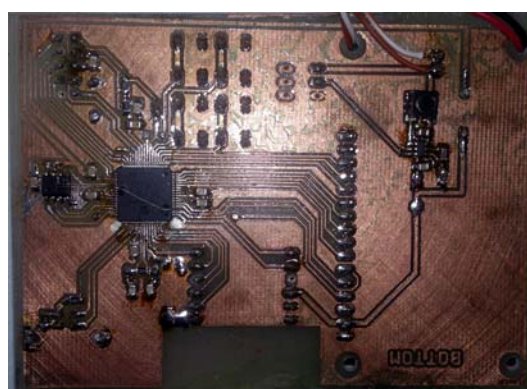
Modul neodesílá žádné zprávy.



Obrázek 16: Plošný spoj modulu s LCD displayem



(a) Pohled na stranu součástek



(b) Pohled na stranu plošného spoje

Obrázek 17: Hotový modul LCD

6 Závěr

Cílem bakalářské práce bylo vyvinout prototyp bezdrátového modulu „rozhraní iDům – ZigBee“, který by zprostředkovával komunikaci mezi systémem inteligentního domu iDům a bezdrátovými moduly, založenými na technologii 802.15.4, tzv. ZigBee. Uvedený modul se podařilo navrhnout a je plně funkční.

Dalším cílem bylo vyvinout prototypy bezdrátových bateriově napájených modulů, které by byly pomocí modulu rozhraní iDům – ZigBee připojeny do systému inteligentního domu iDům. Podařilo se úspěšně vyrobit následující moduly:

- Modul vypínač – plně funkční, odesílá zprávu o změně stavu spínačů,
- Modul LCD – plně funkční, přijímá zprávu, kterou následně zobrazí na LCD panelu. Při manipulaci s modulem se mi nedopatřením podařilo zlikvidovat jeden pin mikrořadice PIC24FJ256GB110, dalo docela práci nalézt zdroj problému, ale nakonec se podařila závada odstranit použitím jiného volného pinu na mikrořadiči.

Modul s analogovým čidlem byl vyvíjen pouze na vývojovém kitu Explorer 16 firmy Microchip.

Plošné spoje byly navrženy v programu Eagle od firmy CadSoft. Následná výroba desek plošných spojů probíhala kompletně v domácích podmínkách. Pokud by výše uvedené moduly měly být použity v praxi, bylo by třeba optimalizovat desky plošných spojů a nechat si je vyrobit, případně i osadit nějakou odbornou firmou, která se tímto zabývá.

7 Reference

- [1] *Bezdrátová komunikace* [online]. [cit. 2011-5-5]. Dostupný z WWW: <<http://www.microrisc.cz/new/webcz/index.php?id=reseni&sekce=komunikace>>.
- [2] BIES, Lammert. *RS232 serial null modem cable wiring* [online]. c2010 [cit. 2011-5-5]. Dostupný z WWW:<http://www.lammertbies.nl/comm/info/RS-232_null_modem.html>.
- [3] *Bluetooth* [online]. 4.5.2011 [cit. 2011-5-5] . Dostupný z WWW: <<http://cs.wikipedia.org/wiki/Bluetooth>>.
- [4] Firemní dokumentace systému iDům.
- [5] FLOWERS, David a YANG, Yifeng. *Microchip MiWi™ Wireless Networking Protocol Stack* [online]. c2010 [cit. 2011-5-5]. Dostupný z WWW: <<http://ww1.microchip.com/downloads/en/AppNotes/AN1066%20-%20MiWi%20App%20Note.pdf>>.
- [6] *IQRF brochure* [online]. [cit. 2011-5-5] . Dostupný z WWW: <http://www.iqrf.org/weben/edit_support/downloads.php?s=BRIQRF_101025.pdf>.
- [7] JASIO, Lucio Di. *Programming 16-Bit PIC Microcontrollers in C Learning to Fly the PIC24* . USA : Elsevier Inc., 2007. ISBN-10: 0-7506-8292-2.
- [8] LATTIBEAUDIERE, Derrick P. *Microchip ZigBee-2006 Residential Stack Protocol* [online]. c2008 [cit. 2011-5-5]. Dostupný z WWW: <<http://ww1.microchip.com/downloads/en/AppNotes/ZigBee2006%20Application%20Note%20AN1232A.pdf>>.
- [9] *MiWi™ DEVELOPMENT ENVIRONMENT* [online]. c2011 [cit. 2011-5-5]. Dostupný z WWW: <http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2664¶m=en520414>.
- [10] *MRF24J40MA Data Sheet* [online]. c2008 [cit. 2011-5-5]. Dostupný z WWW: <<http://ww1.microchip.com/downloads/en/DeviceDoc/70329b.pdf>>.
- [11] *PIC24FJ128GA010 Family Data Sheet* [online]. c2009 [cit. 2011-5-5]. Dostupný z WWW: <<http://ww1.microchip.com/downloads/en/DeviceDoc/39747e.pdf>>.
- [12] *PIC24FJ256GB110 Family Data Sheet* [online]. c2009 [cit. 2011-5-5]. Dostupný z WWW: <<http://ww1.microchip.com/downloads/en/DeviceDoc/39897c.pdf>>.
- [13] VOJÁČEK, Antonín. *ZigBee - novinka na poli bezdrátové komunikace* [online]. 8.6.2005 [cit. 2011-5-5]. Dostupný z WWW: <<http://hw.cz/Rozhrani/ART1299-ZigBee---novinka-na-poli-bezdratove-komunikace.html>>.

- [14] *Wi-Fi* [online]. 25.4.2011 [cit. 2011-5-5]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/Wi-Fi>>.
- [15] *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)* [online]. 1.10.2003 [cit. 2011-5-5]. Dostupný z WWW: <<http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf>>.
- [16] *ZigBee* [online]. 13.4.2010 [cit. 2011-5-5]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/ZigBee>>.
- [17] *256K SPI Bus Serial EEPROM* [online]. c2007 [cit. 2011-5-5]. Dostupný z WWW: <<http://ww1.microchip.com/downloads/en/DeviceDoc/21822F.pdf>>.

A Výpisy kódů

```

void ProcessNONZigBeeTasks(void)
{
    int kkk;
    signed int adr;
    BYTE adri;
    BYTE delka = MaxBufLen;

    CLRWDT();

    //Zprava urcena pro nas
    if (myiDevInfo.zpravaPrijata)
    {
        if (myiDevInfo.zprava.Data.Komu == myiDevInfo.devID) //Zprava pro nas
        {
            switch(myiDevInfo.zprava.Data.Typ)
            {
                case msgByte:
                {
                    VysliAccept();
                    kkk=myiDevInfo.zprava.Data.Obsah.ZByte.Value;
                    ConsolePutROMString( (ROM char *)"Idum_prisla_zprava_typu_msgByte:\r\n" );
                    break;
                }
                case msgArray:
                {
                    VysliAccept();
                    ConsolePutROMString( (ROM char *)"Idum_prisla_zprava_typu_msgArray:\r\n" );
                    break;
                }
                case msgQuestion:
                {
                    VysliAccept();
                    ConsolePutROMString( (ROM char *)"Idum_prisla_zprava_typu_msgQuestion:\r\n" );
                    ;
                    break;
                }
                default:
                {
                    ConsolePutROMString( (ROM char *)"Idum_prisla_zprava_jineho_typu:" );
                    PrintChar( myiDevInfo.zprava.Data.Typ );
                    ConsolePutROMString( (ROM char *)"\r\n" );
                } //END switch(Zprava.Data.Typ)
            }
        }
        else //zprava pro zigbee
        { //otestujeme jestli zname takove zarizeni a preposleme zpravu do jeho bufferu
            signed char adr = FindiDev (myiDevInfo.docasna.Data.Komu);
            if (adr > ERR)
            {
                if (myiDevInfo.docasna.Data.Typ != msgAccept)
                {
                    BYTE tmp;

```

```

        tmp=myiDevInfo.docasna.Data.Delka;
        while(tmp-->0) wirelessDevInfo.data[(BYTE)adr].vysilani.QuickAccess[tmp]=
            myiDevInfo.docasna.QuickAccess[tmp];
        wirelessDevInfo.devState[(BYTE)adr].zpravaiPrijata=TRUE;
        VysliAcceptt(myiDevInfo.docasna.Data.Komu);
    }
}
}
myiDevInfo.zpravaPrijata=FALSE; // zprava byla zpracovana muzeme ji zahodit
}
}

```

Výpis 7: Zpracování přijaté zprávy od systému iDům

```

//kontola jestli neprišla zprava od nejakeho zigbee zarizeni pokud ano odesleme do idum
if (CheckwMSG())//mame zpravu
{
    adr = FindwMSG();
    if (adr == ERR)
    {
        ConsolePutROMString( (ROM char *) "chyba..FindwMSG\r\n" );
    }
    else
    {
        ConsolePutROMString( (ROM char *) "presun_dat_pro_idum:_" );
        adri = (BYTE)adr;
        for (i = 0; i < delka; i++) // zarizeni nalezeno prekopirujeme data do vysilaciho bufferu
            idum
        {
            myiDevInfo.vysilani.QuickAccess[i] = wirelessDevInfo.data[adri].zprava.QuickAccess[i];
            PrintChar( myiDevInfo.vysilani.QuickAccess[i] );

            if (i == poz_DelkaZpr) //prisla delka zpravy;
            {
                delka = myiDevInfo.vysilani.QuickAccess[i];
            }
            ConsolePutROMString( (ROM char *) "_" );
        }
        ConsolePutROMString( (ROM char *) "\r\n" );

        //Zrusime priznak o prijeti zpravy
        wirelessDevInfo.devState[adri].zpravawPrijata = FALSE;
        //Odesleme zpravu
        VysliPole();
    }
}
}

```

Výpis 8: Zpracování přijaté zprávy od bezdrátového zařízení

```

#ifndef _ZIGBEE_DEF
#define _ZIGBEE_DEF

```

```
// PIC Information
#define CLOCK_FREQ 8000000ul
#define BAUD_RATE 19200ul
#define MAX_HEAP_SIZE 2048

// SPI and Non-volatile Storage Information
#define USE_EXTERNAL_NVM
#define EXTERNAL_NVM_BYTES 32768
#define EEPROM_PAGE_SIZE 64
#define EEPROM_nCS LATDbits.LATD12
#define EEPROM_nCS_TRIS TRISDbits.TRISD12
// RF is on SPI1, EE is on SPI2
#define RF_SSPIF_BIT IFS0bits.SPI1IF
#define RF_SSPBUF_REG SPI1BUF
#define EE_SSPIF_BIT IFS2bits.SPI2IF
#define EE_SSPBUF_REG SPI2BUF

// Device MAC Address
#define MAC_LONG_ADDR_BYTE7 0x00
#define MAC_LONG_ADDR_BYTE6 0x00
#define MAC_LONG_ADDR_BYTE5 0x00
#define MAC_LONG_ADDR_BYTE4 0x00
#define MAC_LONG_ADDR_BYTE3 0x00
#define MAC_LONG_ADDR_BYTE2 0x00
#define MAC_LONG_ADDR_BYTE1 0x00
#define MAC_LONG_ADDR_BYTE0 0x02

// ZigBee Device Information
#define I_AM_END_DEVICE
#define I_AM_RFD
#define INCLUDE_ED_SCAN
#define INCLUDE_ACTIVE_SCAN
#define MY_CAPABILITY_INFO 0x80
#define RFD_POLL_RATE (ONE_SECOND * 4)

// ZDO Configuration
#define SUPPORT_END_DEVICE_BINDING
#define CONFIG_ENDDEV_BIND_TIMEOUT (ONE_SECOND * 5)

// APS Configuration
#define MAX_APL_FRAMES 10
#define MAX_APS_ACK_FRAMES 10
#define MAX_APS_ADDRESSES 10
#define I_SUPPORT_BINDINGS
#define NUM_BUFFERED_INDIRECT_MESSAGES 0
#define MAX_BINDINGS 25
#define MAX_DUPLICATE_TABLE 21
#define DUPLICATE_TABLE_EXPIRATION 5
#define I_SUPPORT_GROUP_ADDRESSING
#define MAX_GROUP 8
#define MAX_GROUP_END_POINT 8
#define MAX_GROUP_RECORD_BUFFER 5

// NWK Configuration
```

```

#define MAX_NEIGHBORS 9
#define NUM_BUFFERED_BROADCAST_MESSAGES 0
#define NUM_BUFFERED_ROUTING_MESSAGES 0

// MAC Configuration
#define MINIMUM_JOIN_LQI 48
#define RX_BUFFER_SIZE 256
#define RX_BUFFER_LOCATION 0x400
#define MAC_PIB_macBeaconOrder 15
#define MAC_PIB_macSuperframeOrder 15
#define MAC_PIB_macBattLifeExt FALSE

// Transceiver Information
#define RF_CHIP MRF24J40
#define MAC_PIB_macAckWaitDuration (54+3)
#define RF_INT_PIN PORTEbits.RE8
#define RFIF_IFS1bits.INT1IF
#define RFIE_IEC1bits.INT1IE
#define PHY_CS_LATBbits.LATB2
#define PHY_CS_TRIS_TRISBbits.TRISB2
#define PHY_RESETn_LATGbits.LATG2
#define PHY_RESETn_TRIS_TRISGbits.TRISG2
#define PHY_WAKE_LATGbits.LATG3
#define PHY_WAKE_TRIS_TRISGbits.TRISG3
#define PA_LEVEL 0x00 // -0.00 dBm
#define FREQUENCY_BAND FB_2400GHz
#define ALLOWED_CHANNELS 0x04000000
// SPI information is defined with the non-volatile storage information

// Profile and Endpoint Configuration
#define NUM_USER_ENDPOINTS 2
#define Src_EndPoint 1
#define Dst_EndPoint 240
#define MY_MAX_INPUT_CLUSTERS 14
#define MY_MAX_OUTPUT_CLUSTERS 14
#include "..\ZigBeeStack\zTest.h"

#endif

```

Výpis 9: Ukázka souboru zigbee.def

```

#include "zigbee.def"
#include "zNWK.h"
#include "zZDO.h"

ROM NODE_DESCRIPTOR Config_Node_Descriptor =
{
    0x02,           // ZigBee End Device
    0x00,           // (reserved)
    0x00,           // (APS Flags, not currently used)
    0x08,           // Frequency Band 2400
    MY_CAPABILITY_INFO, // Capability Information
    {0x00, 0x00},   // Manufacturer Code
    0x7F,           // Max Buffer Size

```

```

    {0x00, 0x00}      // Max Transfer Size
};

ROM NODE_POWER_DESCRIPTOR Config_Power_Descriptor =
{
    0x00, //Power mode: RxOn
    0x05, //Available power: Mains DispBatt
    0x01, //Current power: Mains
    0x0c // Fill in current power level
};

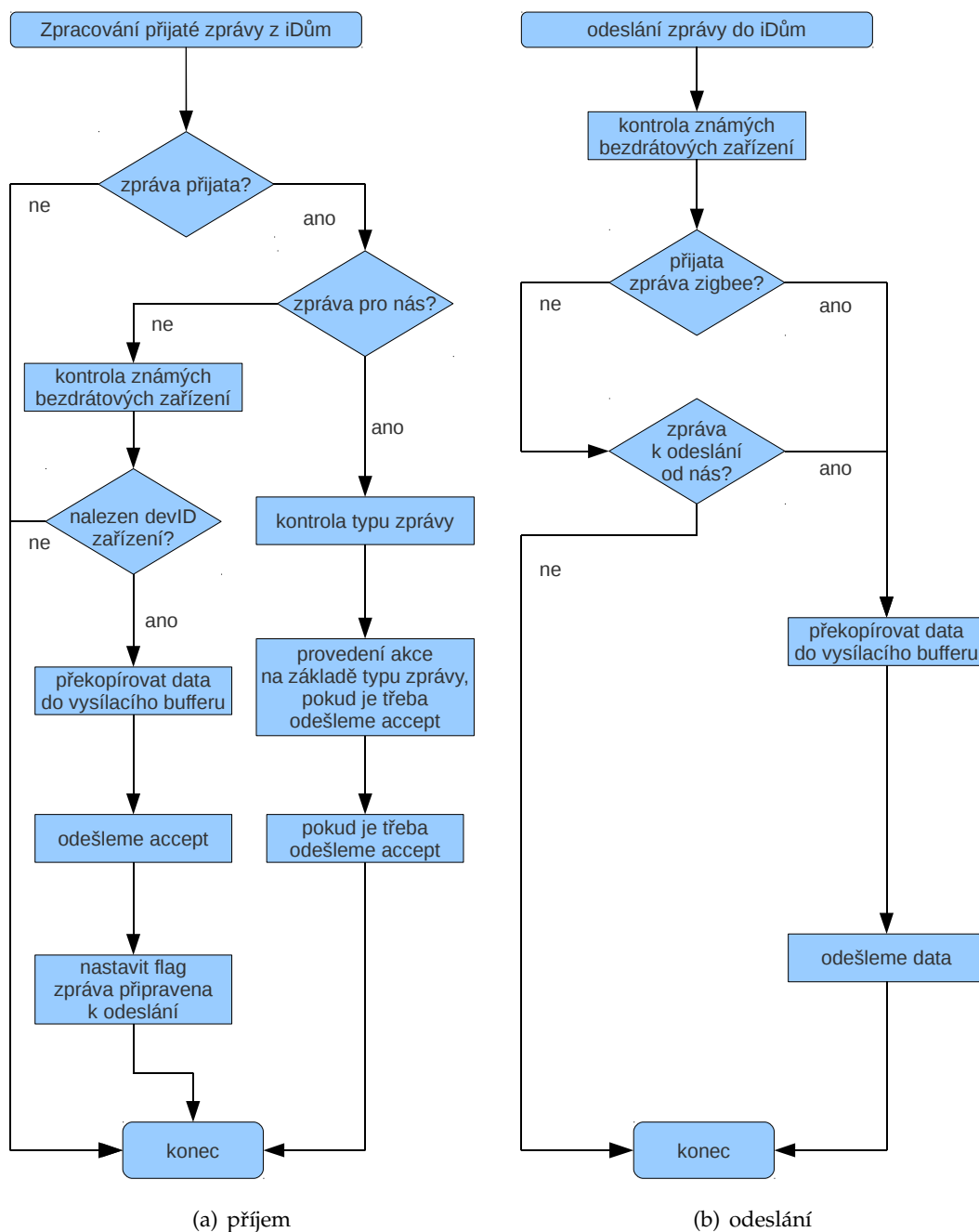
ROM NODE_SIMPLE_DESCRIPTOR Config_Simple_Descriptors[1] =
{
    //-----
    // ZigBee Device Object Endpoint
    // DO NOT MODIFY THIS DESCRIPTOR!!!
    //-----
    {
        EP_ZDO,
        {0x00, 0x00}, // ZDO Profile ID
        {0x00, 0x00}, // ZDO Device
        0x00,         // ZDO Version
        NO_OTHER_DESCRIPTOR_AVAILABLE,
        ZDO_INPUT_CLUSTERS,
        { NWK_ADDR_req, IEEE_ADDR_req, NODE_DESC_req, POWER_DESC_req,
          SIMPLE_DESC_req, ACTIVE_EP_req, MATCH_DESC_req
        },
        ZDO_OUTPUT_CLUSTERS,
        { NWK_ADDR_rsp, IEEE_ADDR_rsp, NODE_DESC_rsp, POWER_DESC_rsp,
          SIMPLE_DESC_rsp, ACTIVE_EP_rsp, MATCH_DESC_rsp
        },
    }
};

ROM _Config_NWK_Mode_and_Params Config_NWK_Mode_and_Params =
{
    nwkcProtocolVersion, //Protocol Version
    MY_STACK_PROFILE_ID, //Stack Profile ID
    MAC_PIB_macBeaconOrder, //Beacon Order
    MAC_PIB_macSuperframeOrder, //Superframe Order
    MAC_PIB_macBattLifeExt, //Battery Life Extension
    PROFILE_nwkSecurityLevel, //Security Level
    ALLOWED_CHANNELS //Channels to scan
};

```

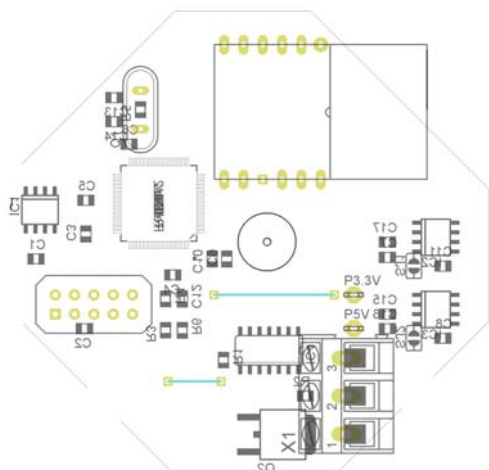
Výpis 10: Ukázka souboru myZigBee.c

B Vývojové diagramy

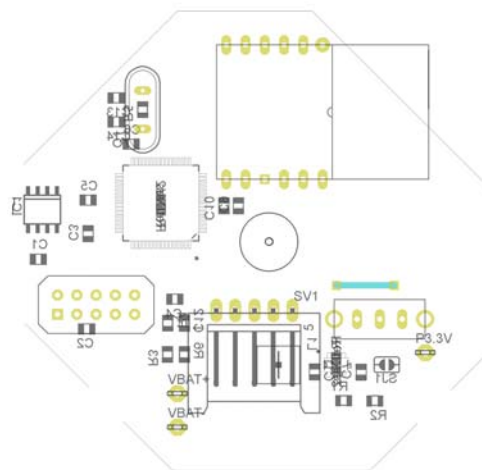


Obrázek 18: Zpracování iDům zprávy

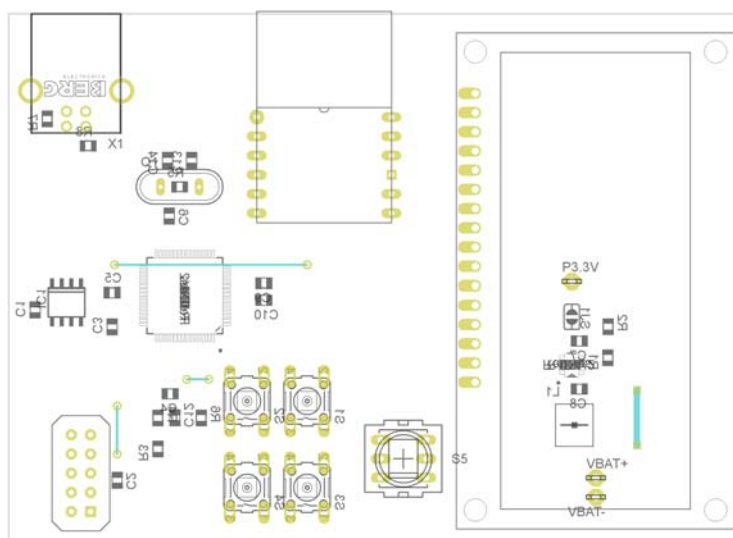
C Osazovací plányky



(a) modul rozhraní iDum – ZigBee



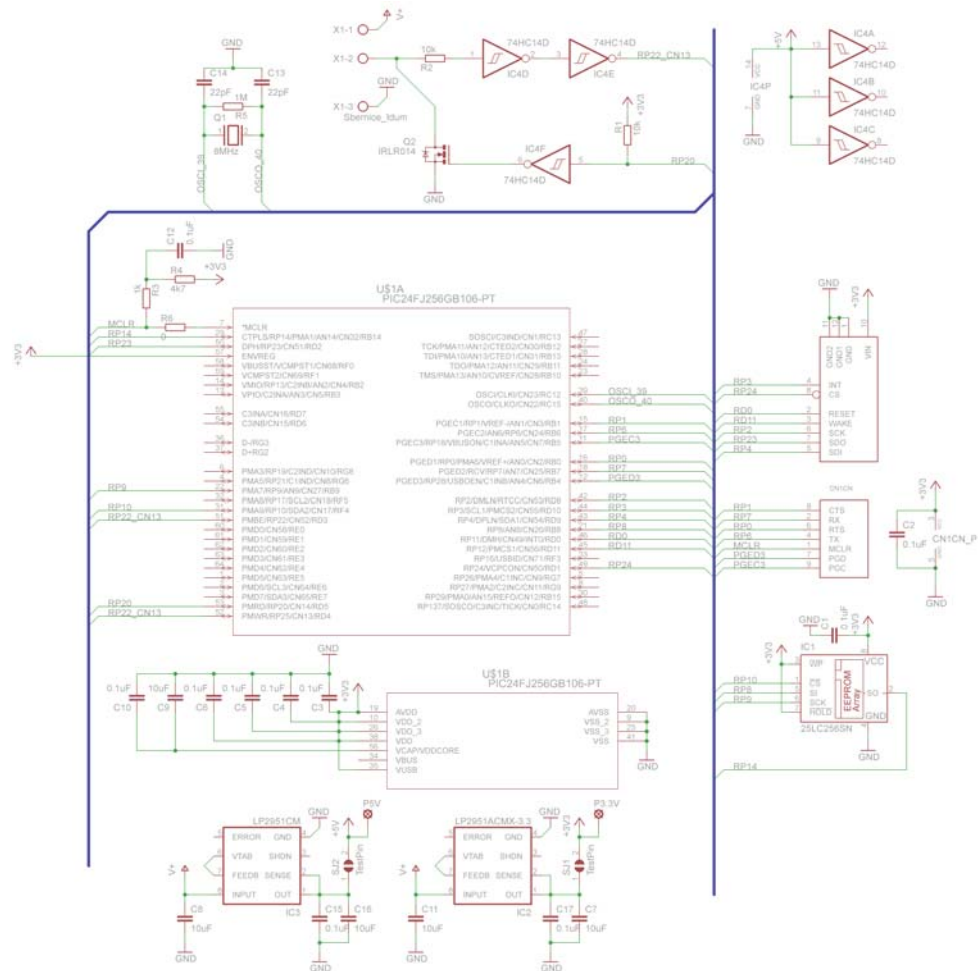
(b) Osazovací plán



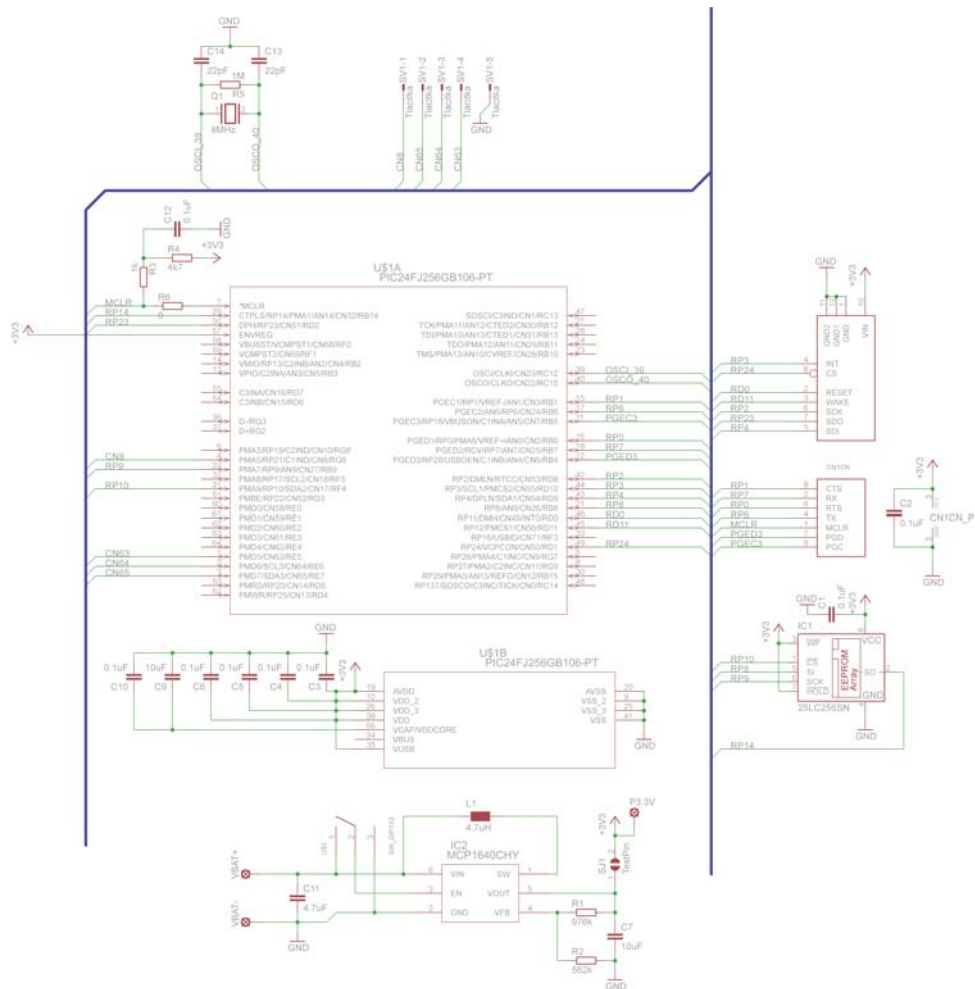
(c) modul LCD

Obrázek 19: Osazovací plány modulů

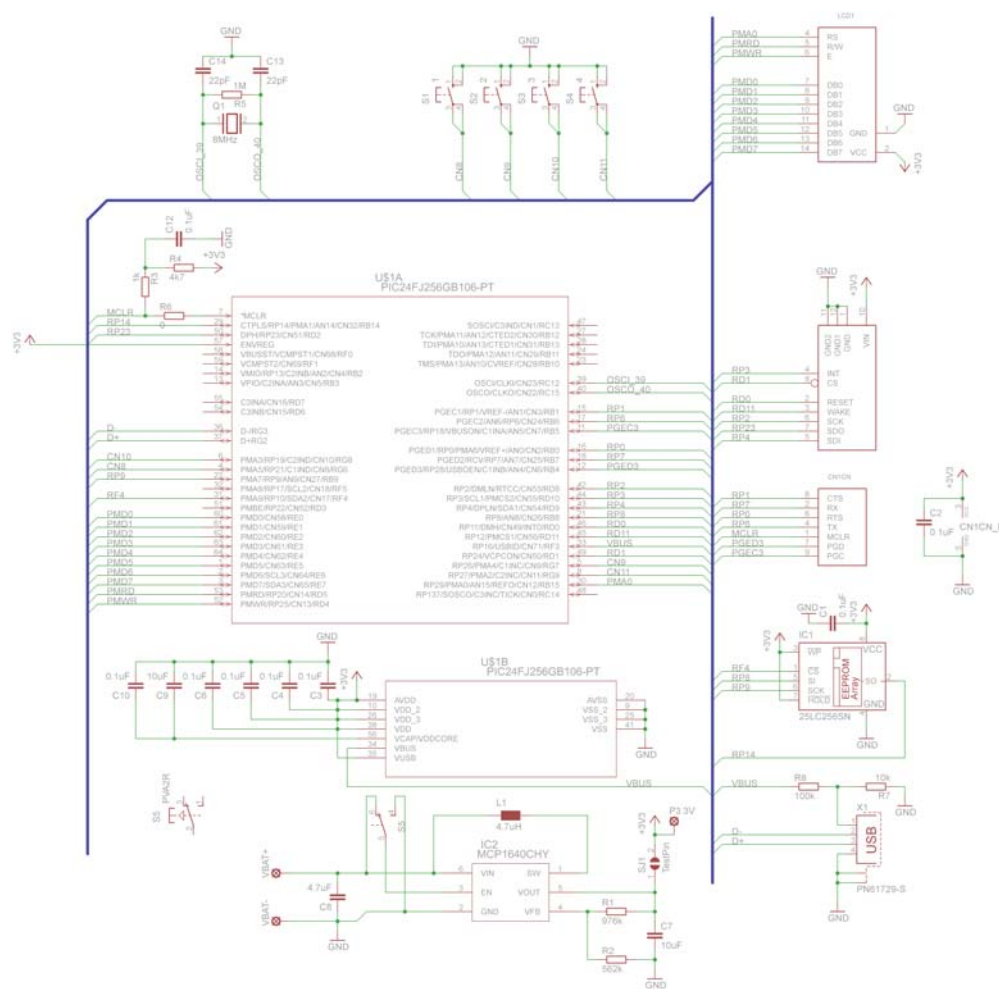
D Schémata zapojení



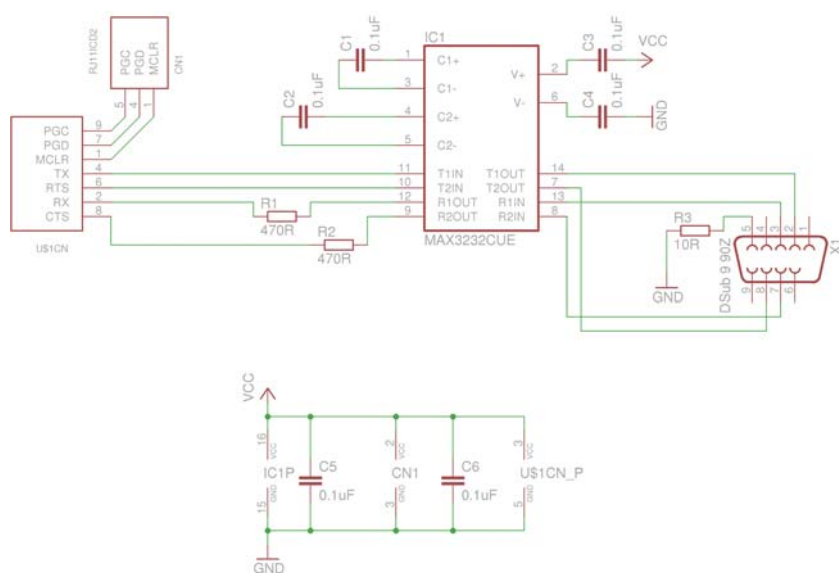
Obrázek 20: Schéma modulu rozhraní iDum – ZigBee



Obrázek 21: Schéma modulu vypínač



Obrázek 22: Schéma modulu s LCD displayem



Obrázek 23: Schéma propojovacího modulu